Burroughs

B 6700

HANDBOOK

VOLUME I

HARDWARE

THIS HANDBOOK REPLACES FORM NO. 5000276
DATED JULY 1971

GENERAL INFORMATION

PROCESSOR OPERATORS

MDL OPERATORS

DCP INSTRUCTIONS

WORD FORMATS

I/O DESCRIPTOR FORMATS

HARDWARE INTERRUPTS

MISCELLANEOUS HARDWARE INFORMATION

MAINTENANCE INFORMATION

$7.00 (VOLUMES I AND II)

PRINTED IN U.S. AMERICA 1-72 5000276

LIST OF EFFECTIVE PAGES

NOTE: Insert latest changed pages; dispose of superseded pages.

TOTAL NUMBER OF PAGES IN VOLUME I IS 386, CONSISTING OF THE FOLLOWING:

Sections 9 thru 15 are in Volume II.

VOLUME I

TABLE OF CONTENTS

PRINTED IN U.S. AMERICA 5000276

TABLE OF CONTENTS (CONT)

PRINTED IN U.S. AMERICA 5000276

TABLE OF CONTENTS (CONT)

TABLE OF CONTENTS (CONT)

TABLE OF CONTENTS (CONT)

TABLE OF CONTENTS (CONT)

PRINTED IN U.S. AMERICA 5000276

GENERAL INFORMATION

The terms "Multiplexor" and "I/O Processor" are synonymous.

MULTIPLEXOR CONFIGURATION CHART

_____ System Located at _____ Page _____ of _____

PCC CAB			CHAN. ASSIGN					GROUP ASSIGN					RDY. STS. VEC.		
			MPX PRIOR-	DBFH6/ FIO		DBHH8/ FHB		MINTERM	JUMPERS	REQUIRED	GROUP PAD	CHAN PAD	LOCATION		
CAB	CON	TYPE	ITY	ARL	BUSY	CDL	AGL	GROUP	DBHF2/F14	DBHF0/F16	DBF12	DBHF4	PCC/#	EXCHANGE	MPX
	0			1		2									
	1			3		4									
	2			5		6									
	3			7		8									
	4			9		10									
	0			11		12									
	1			13		14									
	2			15		16									
	3			17		18									
	4			19		20									
	5			21		22									
	6			23		24									
	7			25		26									
	8			27		28									
	9			29		30									
	5			31		32									
	6			33		34									
	7			35		36									
	8			37		38									
	9			39		40									

Refer to section 4 for configuration instructions.

PRINTED IN U.S. AMERICA

5000276

MULTIPLEXOR CONFIGURATION CHART

_____ System Located at _____ Page _____ of _____

PCC CAB			CHAN. ASSIGN					GROUP ASSIGN					RDY. STS. VEC.		
			MPX PRIOR-	DBFH6/ FIO		DBHH8/ FHB		MINTERM	JUMPERS	REQUIRED	GROUP PAD	CHAN PAD	LOCATION		
CAB	CON	TYPE	ITY	ARL	BUSY	CDL	AGL	GROUP	DBHF2/F14	DBHF0/F16	DBF12	DBHF4	PCC/#	EXCHANGE	MPX
	0			1		2									
	1			3		4									
	2			5		6									
	3			7		8									
	4			9		10									
	0			11		12									
	1			13		14									
	2			15		16									
	3			17		18									
	4			19		20									
	5			21		22									
	6			23		24									
	7			25		26									
	8			27		28									
	9			29		30									
	5			31		32									
	6			33		34									
	7			35		36									
	8			37		38									
	9			39		40									

PRINTED IN U.S. AMERICA

5000276

MULTIPLEXOR CONFIGURATION CHART

_____ System Located at _____ Page _____ of _____

PCC CAB			CHAN. ASSIGN					GROUP ASSIGN					RDY. STS. VEC.		
			MPX	DBFH6/ F10		DBHH8/ FHB							LOCATION		
CAB	CON	TYPE	PRIORITY	ARL	BUSY	CDL	AGL	MINTERM GROUP	JUMPERS DBHF2/F14	REQUIRED DBHF0/F16	GROUP PAD DBF12	CHAN PAD DBHF4	PCC/#	EXCHANGE	MPX
	0			1		2									
	1			3		4									
	2			5		6									
	3			7		8									
	4			9		10									
	0			11		12									
	1			13		14									
	2			15		16									
	3			17		18									
	4			19		20									
	5			21		22									
	6			23		24									
	7			25		26									
	8			27		28									
	9			29		30									
	5			31		32									
	6			33		34									
	7			35		36									
	8			37		38									
	9			39		40									

PRINTED IN U.S. AMERICA 5000276

MULTIPLEXOR CONFIGURATION CHART

_____ System Located at _____ Page _____ of _____

PCC CAB			CHAN. ASSIGN				GROUP ASSIGN					RDY. STS. VEC.		
			MPX PRIOR	DBFH6/ F10		DBHH8/ FHB						LOCATION		
CAB	CON	TYPE	ITY	ARL	BUSY	CDL	AGL	MINTERM GROUP	JUMPERS DBHF2/F14	REQUIRED DBHF0/F16	GROUP PAD DBF12	CHAN PAD DBHF4	PCC/#	EXCHANGE	MPX
	0			1		2									
	1			3		4									
	2			5		6									
	3			7		8									
	4			9		10									
	0			11		12									
	1			13		14									
	2			15		16									
	3			17		18									
	4			19		20									
	5			21		22									
	6			23		24									
	7			25		26									
	8			27		28									
	9			29		30									
	5			31		32									
	6			33		34									
	7			35		36									
	8			37		38									
	9			39		40									

PRINTED IN U.S. AMERICA
5000276

MULTIPLEXOR CONFIGURATION CHART

_____ System Located at _____ Page _____ of _____

PCC CAB			CHAN. ASSIGN					GROUP ASSIGN					RDY. STS. VEC.		
			MPX PRIOR-	DBFH6/ F10		DBHH8/ FHB		MINTERM	JUMPERS	REQUIRED	GROUP PAD	CHAN PAD	LOCATION		
CAB	CON	TYPE	ITY	ARL	BUSY	CDL	AGL	GROUP	DBHF2/F14	DBHFO/F16	DBF12	DBHF4	PCC/#	EXCHANGE	MPX
	0			1		2									
	1			3		4									
	2			5		6									
	3			7		8									
	4			9		10									
	0			11		12									
	1			13		14									
	2			15		16									
	3			17		18									
	4			19		20									
	5			21		22									
	6			23		24									
	7			25		26									
	8			27		28									
	9			29		30									
	5			31		32									
	6			33		34									
	7			35		36									
	8			37		38									
	9			39		40									

PRINTED IN U.S. AMERICA

5000276

MULTIPLEXOR CONFIGURATION CHART

_____ System Located at _____ Page _____ of _____

PCC CAB			CHAN. ASSIGN					GROUP ASSIGN					RDY. STS. VEC.		
			MPX	DBFH6/ F10		DBHH8/ FHB					GROUP	CHAN	LOCATION		
CAB	CON	TYPE	PRIORITY	ARL	BUSY	CDL	AGL	MINTERM GROUP	JUMPERS DBHF2/F14	REQUIRED DBHF0/F16	PAD DBF12	PAD DBHF4	PCC/#	EXCHANGE	MPX
	0			1		2									
	1			3		4									
	2			5		6									
	3			7		8									
	4			9		10									
	0			11		12									
	1			13		14									
	2			15		16									
	3			17		18									
	4			19		20									
	5			21		22									
	6			23		24									
	7			25		26									
	8			27		28									
	9			29		30									
	5			31		32									
	6			33		34									
	7			35		36									
	8			37		38									
	9			39		40									

PRINTED IN U.S. AMERICA

5000276

MULTIPLEXOR CONFIGURATION CHART

_____ System Located at _____ Page _____ of _____

PCC CAB			CHAN. ASSIGN				GROUP ASSIGN					RDY. STS. VEC.		
			MPX PRIOR-	DBFH6/ F10		DBHH8/ FHB						LOCATION		
CAB	CON	TYPE	ITY	ARL	BUSY	CDL	AGL	MINTERM GROUP	JUMPERS DBHF2/F14	REQUIRED DBHF0/F16	GROUP PAD DBF12	CHAN PAD DBHF4	PCC/#	EXCHANGE	MPX
	0			1		2									
	1			3		4									
	2			5		6									
	3			7		8									
	4			9		10									
	0			11		12									
	1			13		14									
	2			15		16									
	3			17		18									
	4			19		20									
	5			21		22									
	6			23		24									
	7			25		26									
	8			27		28									
	9			29		30									
	5			31		32									
	6			33		34									
	7			35		36									
	8			37		38									
	9			39		40									

PRINTED IN U.S. AMERICA 5000276

MULTIPLEXOR CONFIGURATION CHART

_____ System Located at _____ Page _____ of _____

PCC CAB			CHAN. ASSIGN				GROUP ASSIGN					RDY. STS. VEC.		
			MPX	DBFH6/ F10		DBHH8/ FHB				GROUP	CHAN	LOCATION		
CAB	CON	TYPE	PRIOR-ITY	ARL	BUSY	CDL	AGL	MINTERM GROUP	JUMPERS DBHF2/F14	REQUIRED DBHF0/F16	PAD DBF12	PAD DBHF4	PCC/#	EXCHANGE	MPX
	0			1		2									
	1			3		4									
	2			5		6									
	3			7		8									
	4			9		10									
	0			11		12									
	1			13		14									
	2			15		16									
	3			17		18									
	4			19		20									
	5			21		22									
	6			23		24									
	7			25		26									
	8			27		28									
	9			29		30									
	5			31		32									
	6			33		34									
	7			35		36									
	8			37		38									
	9			39		40									

PRINTED IN U.S. AMERICA

5000276

MULTIPLEXOR CONFIGURATION CHART

_____ System Located at _____ Page _____ of _____

PCC CAB			CHAN. ASSIGN				GROUP ASSIGN					RDY. STS. VEC.		
			MPX	DBFH6/ F10		DBHH8/ FHB				GROUP	CHAN	LOCATION		
CAB	CON	TYPE	PRIOR-ITY	ARL	BUSY	CDL	AGL	MINTERM GROUP	JUMPERS DBHF2/F14	REQUIRED DBHF0/F16	PAD DBF12	PAD DBHF4	PCC/#	EXCHANGE	MPX
	0			1		2									
	1			3		4									
	2			5		6									
	3			7		8									
	4			9		10									
	0			11		12									
	1			13		14									
	2			15		16									
	3			17		18									
	4			19		20									
	5			21		22									
	6			23		24									
	7			25		26									
	8			27		28									
	9			29		30									
	5			31		32									
	6			33		34									
	7			35		36									
	8			37		38									
	9			39		40									

PRINTED IN U.S. AMERICA
5000276

MULTIPLEXOR CONFIGURATION CHART

_____ System Located at _____ Page _____ of _____

PCC CAB			CHAN. ASSIGN				GROUP ASSIGN					RDY. STS. VEC.		
			MPX PRIOR-	DBFH6/ FIO		DBHH8/ FHB				GROUP PAD	CHAN PAD	LOCATION		
CAB	CON	TYPE	ITY	ARL	BUSY	CDL	AGL	MINTERM GROUP	JUMPERS DBHF2/F14	REQUIRED DBHF0/F16	DBF12	DBHF4	PCC/#	EXCHANGE	MPX
	0			1		2									
	1			3		4									
	2			5		6									
	3			7		8									
	4			9		10									
	0			11		12									
	1			13		14									
	2			15		16									
	3			17		18									
	4			19		20									
	5			21		22									
	6			23		24									
	7			25		26									
	8			27		28									
	9			29		30									
	5			31		32									
	6			33		34									
	7			35		36									
	8			37		38									
	9			39		40									

PRINTED IN U.S. AMERICA

5000276

MULTIPLEXOR CONFIGURATION CHART

_____ System Located at _____ Page _____ of _____

PCC CAB			CHAN. ASSIGN				GROUP ASSIGN					RDY. STS. VEC.		
			MPX PRIOR-	DBFH6/ F10		DBHH8/ FHB				GROUP PAD	CHAN PAD	LOCATION		
CAB	CON	TYPE	ITY	ARL	BUSY	CDL	AGL	MINTERM GROUP	JUMPERS DBHF2/F14	REQUIRED DBHF0/F16	DBF12	DBHF4	PCC/#	EXCHANGE	MPX
	0			1		2									
	1			3		4									
	2			5		6									
	3			7		8									
	4			9		10									
	0			11		12									
	1			13		14									
	2			15		16									
	3			17		18									
	4			19		20									
	5			21		22									
	6			23		24									
	7			25		26									
	8			27		28									
	9			29		30									
	5			31		32									
	6			33		34									
	7			35		36									
	8			37		38									
	9			39		40									

MULTIPLEXOR CONFIGURATION CHART

_____ System Located at _____ Page _____ of _____

PCC CAB			CHAN. ASSIGN				GROUP ASSIGN					RDY. STS. VEC.		
			MPX PRIOR ITY	DBFH6/ F10		DBHH8/ FHB						LOCATION		
CAB	CON	TYPE		ARL	BUSY	CDL	AGL	MINTERM GROUP	JUMPERS DBHF2/F14	REQUIRED DBHF0/F16	GROUP PAD DBF12	CHAN PAD DBHF4	PCC/#	EXCHANGE	MPX
	0			1		2									
	1			3		4									
	2			5		6									
	3			7		8									
	4			9		10									
	0			11		12									
	1			13		14									
	2			15		16									
	3			17		18									
	4			19		20									
	5			21		22									
	6			23		24									
	7			25		26									
	8			27		28									
	9			29		30									
	5			31		32									
	6			33		34									
	7			35		36									
	8			37		38									
	9			39		40									

PRINTED IN U.S. AMERICA

5000276

NUMBERING SYSTEMS

Order of Magnitude Table

REGISTER BIT SET	DECIMAL	DECIMAL RECIPROCAL	HEX.	OCTAL	BINARY
0	1	1.0	16^0	8^0	2^0
1	2	0.5			
2	4	0.25			
3	8	0.125		8^1	2^3
4	16	0.062 5	16^1		
5	32	0.031 25			
6	64	0.015 625		8^2	2^6
7	128	0.007 812 5			
8	256	0.003 906 25	16^2		
9	512	0.001 953 125		8^3	2^9
10	1 024	0.000 976 562 5			
11	2 048	0.000 488 281 25			
12	4 096	0.000 244 140 625	16^3	8^4	2^{12}
13	8 192				
14	16 384				
15	32 768			8^5	2^{15}
16	65 536		16^4		
17	131 072				
18	262 144			8^6	2^{18}
19	524 288				
20	1 048 576		16^5		
21	2 097 152			8^7	2^{21}
22	4 194 304				
23	8 388 608				
24	16 777 216		16^6	8^8	2^{24}
25	33 554 432				
26	67 108 864			8^9	2^{27}
27	134 217 728				
28	268 435 456		16^7		
29	536 870 912				
30	1 073 741 824			8^{10}	2^{30}
31	2 147 483 648				
32	4 294 967 296		16^8		
33	8 589 934 592			8^{11}	2^{33}
34	17 179 869 184				
35	34 359 738 368				
36	68 719 476 736		16^9	8^{12}	2^{36}
37	137 438 953 472				
38	274 877 906 944				
*	549 755 813 887				
39	549 755 813 888			8^{13}	2^{39}

* 1st 39 set (Maximum integer value allowed)

NUMBERING SYSTEMS

Hexadecimal-Decimal Conversion Table

x	0x	1x	2x	3x	4x	5x
0	0	65536	131072	196608	262144	327680
1	4096	69632	135168	200704	266240	331776
2	8192	73728	139264	204800	270336	335872
3	12288	77824	143360	208896	274432	339968
4	16384	81920	147456	212992	278528	344064
5	20480	86016	151552	217088	282624	348160
6	24576	90112	155648	221184	286720	352256
7	28672	94208	159744	225280	290816	356352
8	32768	98304	163840	229376	294912	360448
9	36864	102400	167936	233472	299008	364544
A	40960	106496	172032	237568	303104	368640
B	45056	110592	176128	241664	307200	372736
C	49152	114688	180224	245760	311296	376832
D	53248	118784	184320	249856	315392	380928
E	57344	122880	188416	253952	319488	385024
F	61440	126976	192512	258048	323584	389120

x	6x	7x	8x	9x	Ax	Bx
0	393216	458752	524288	589824	655360	720896
1	397312	462848	528384	593920	659456	724992
2	401408	466944	532480	598016	663552	729088
3	405504	471040	536576	602112	667648	733184
4	409600	475136	540672	606208	671744	737280
5	413696	479232	544768	610304	675840	741376
6	417792	483328	548864	614400	679936	745472
7	421888	487424	552960	618496	684032	749568
8	425984	491520	557056	622592	688128	753664
9	430080	495616	561152	626688	692224	757760
A	434176	499712	565248	630784	696320	761856
B	438272	503808	569344	634880	700416	765952
C	442368	507904	573440	638976	704512	770048
D	446464	512000	577536	643072	708608	774144
E	450560	516096	581632	647168	712704	778240
F	454656	520192	585728	651264	716800	782336

x	Cx	Dx	Ex	Fx
0	786432	851968	917504	983040
1	790528	856064	921600	987136
2	794624	860160	925696	991232
3	798720	864256	929792	995328
4	802816	868352	933888	999424
5	806912	872448	937984	1003520
6	811008	876544	942080	1007616
7	815104	880640	946176	1011712
8	819200	884736	950272	1015808
9	823296	888832	954368	1019904
A	827392	892928	958464	1024000
B	831488	897024	962560	1028096
C	835584	901120	966656	1032192
D	839680	905216	970752	1036288
E	843776	909312	974848	1040384
F	847872	913408	978944	1044480

The preceding table and the following table provide for direct conversion between hexadecimal and decimal numbers in the range of:

Hexidecimal

00000 to FFFFF

Decimal

0 to 1048575

The preceding table provides the values of the first 2 of a 5-hexadecimal digit number (nn---), and the following table provides the last 3 (--nnn).

HEXADECIMAL-DECIMAL CONVERSION TABLE (Cont)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
000	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
010	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
020	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
030	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
040	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
050	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
060	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
070	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
080	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
090	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
0A0	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
0B0	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
0C0	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
0D0	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
0E0	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
0F0	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
100	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271
110	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287
120	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303
130	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319
140	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335
150	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351
160	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367
170	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383

HEXADECIMAL-DECIMAL CONVERSION TABLE (Cont.)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
180	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399
190	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415
1A0	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431
1B0	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447
1C0	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463
1D0	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479
1E0	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495
1F0	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511
200	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527
210	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543
220	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559
230	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575
240	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591
250	592	593	594	595	596	597	598	599	600	601	602	603	604	605	606	607
260	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623
270	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639
280	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655
290	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671
2A0	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687
2B0	688	689	690	691	692	693	694	695	696	697	698	699	700	701	702	703
2C0	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719
2D0	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735
2E0	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751
2F0	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767

HEXADECIMAL-DECIMAL CONVERSION TABLE (Cont.)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
300	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783
310	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799
320	800	801	802	803	804	805	806	807	808	809	810	811	812	813	814	815
330	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831
340	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847
350	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863
360	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879
370	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895
380	896	897	898	899	900	901	902	903	904	905	906	907	908	909	910	911
390	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927
3A0	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943
3B0	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959
3C0	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975
3D0	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991
3E0	992	993	994	995	996	997	998	999	1000	1001	1002	1003	1004	1005	1006	1007
3F0	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023
400	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039
410	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055
420	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071
430	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087
440	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103
450	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119
460	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135
470	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151

HEXADECIMAL-DECIMAL CONVERSION TABLE (Cont.)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
480	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167
490	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183
4A0	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199
4B0	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215
4C0	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231
4D0	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247
4E0	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263
4F0	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279
500	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295
510	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311
520	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327
530	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343
540	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359
550	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375
560	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391
570	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407
580	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423
590	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439
5A0	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455
5B0	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471
5C0	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487
5D0	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503
5E0	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519
5F0	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535

PRINTED IN U.S. AMERICA 5000276

HEXADECIMAL-DECIMAL CONVERSION TABLE (Cont)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
600	1536	1537	1538	1539	1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551
610	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567
620	1568	1569	1570	1571	1572	1573	1574	1575	1576	1577	1578	1579	1580	1581	1582	1583
630	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
640	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615
650	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
660	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647
670	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	1662	1663
680	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679
690	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695
6A0	1696	1697	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1709	1710	1711
6B0	1712	1713	1714	1715	1716	1717	1718	1719	1720	1721	1722	1723	1724	1725	1726	1727
6C0	1728	1729	1730	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743
6D0	1744	1745	1746	1747	1748	1749	1750	1751	1752	1753	1754	1755	1756	1757	1758	1759
6E0	1760	1761	1762	1763	1764	1765	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775
6F0	1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787	1788	1789	1790	1791
700	1792	1793	1794	1795	1796	1797	1798	1799	1800	1801	1802	1803	1804	1805	1806	1807
710	1808	1809	1810	1811	1812	1813	1814	1815	1816	1817	1818	1819	1820	1821	1822	1823
720	1824	1825	1826	1827	1828	1829	1830	1831	1832	1833	1834	1835	1836	1837	1838	1839
730	1840	1841	1842	1843	1844	1845	1846	1847	1848	1849	1850	1851	1852	1853	1854	1855
740	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871
750	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887
760	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903
770	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919

HEXADECIMAL-DECIMAL CONVERSION TABLE (cont.)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
780	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935
790	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1959	1950	1951
7A0	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967
7B0	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983
7C0	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
7D0	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
7E0	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031
7F0	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047
800	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063
810	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079
820	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095
830	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111
840	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127
850	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143
860	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159
870	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175
880	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191
890	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207
8A0	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223
8B0	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239
8C0	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255
8D0	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271
8E0	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287
8F0	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303

PRINTED IN U.S. AMERICA

5000276

HEXADECIMAL-DECIMAL CONVERSION TABLE (Cont)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
900	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319
910	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335
920	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351
930	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367
940	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383
950	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399
960	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415
970	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431
980	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447
990	2448	2449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463
9A0	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479
9B0	2480	2481	2482	2483	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495
9C0	2496	2497	2498	2499	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509	2510	2511
9D0	2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525	2526	2527
9E0	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543
9F0	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556	2557	2558	2559
A00	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569	2570	2571	2572	2573	2574	2575
A10	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586	2587	2588	2589	2590	2591
A20	2592	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604	2605	2606	2607
A30	2608	2609	2610	2611	2612	2613	2614	2615	2616	2617	2618	2619	2620	2621	2622	2623
A40	2624	2625	2626	2627	2628	2629	2630	2631	2632	2633	2634	2635	2636	2637	2638	2639
A50	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652	2653	2654	2655
A60	2656	2657	2658	2659	2660	2661	2662	2663	2664	2665	2666	2667	2668	2669	2670	2671
A70	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687

PRINTED IN U.S. AMERICA 5000276

HEXADECIMAL-DECIMAL CONVERSION TABLE (Cont)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A80	2688	2689	2690	2691	2692	2693	2694	2695	2696	2697	2698	2699	2700	2701	2702	2703
A90	2704	2705	2706	2707	2708	2709	2710	2711	2712	2713	2714	2715	2716	2717	2718	2719
AA0	2720	2721	2722	2723	2724	2725	2726	2727	2728	2729	2730	2731	2732	2733	2734	2735
AB0	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	2748	2749	2750	2751
AC0	2752	2753	2754	2755	2756	2757	2758	2759	2760	2761	2762	2763	2764	2765	2766	2767
AD0	2768	2769	2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	2780	2781	2782	2783
AE0	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795	2796	2797	2798	2799
AF0	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2813	2814	2815
B00	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825	2826	2827	2828	2829	2830	2831
B10	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842	2843	2844	2845	2846	2847
B20	2848	2849	2850	2851	2852	2853	2854	2855	2856	2857	2858	2859	2860	2861	2862	2863
B30	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879
B40	2880	2881	2882	2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893	2894	2895
B50	2896	2897	2898	2899	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909	2910	2911
B60	2912	2913	2914	2915	2916	2917	2918	2919	2920	2921	2922	2923	2924	2925	2926	2927
B70	2928	2929	2930	2931	2932	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943
B80	2944	2945	2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959
B90	2960	2961	2962	2963	2964	2965	2966	2967	2968	2969	2970	2971	2972	2973	2974	2975
BA0	2976	2977	2978	2979	2980	2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991
BB0	2992	2993	2994	2995	2996	2997	2998	2999	3000	3001	3002	3003	3004	3005	3006	3007
BC0	3008	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020	3021	3022	3023
BD0	3024	3025	3026	3027	3028	3029	3030	3031	3032	3033	3034	3035	3036	3037	3038	3039
BE0	3040	3041	3042	3043	3044	3045	3046	3047	3048	3049	3050	3051	3052	3053	3054	3055
BF0	3056	3057	3058	3059	3060	3061	3062	3063	3064	3065	3066	3067	3068	3069	3070	3071

HEXADECIMAL-DECIMAL CONVERSION TABLE (Cont.)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
C00	3072	3073	3074	3075	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3086	3087
C10	3088	3089	3090	3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102	3103
C20	3104	3105	3106	3107	3108	3109	3110	3111	3112	3113	3114	3115	3116	3117	3118	3119
C30	3120	3121	3122	3123	3124	3125	3126	3127	3128	3129	3130	3131	3132	3133	3134	3135
C40	3136	3137	3138	3139	3140	3141	3142	3143	3144	3145	3146	3147	3148	3149	3150	3151
C50	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	3163	3164	3165	3166	3167
C60	3168	3169	3170	3171	3172	3173	3174	3175	3176	3177	3178	3179	3180	3181	3182	3183
C70	3184	3185	3186	3187	3188	3189	3190	3191	3192	3193	3194	3195	3196	3197	3198	3199
C80	3200	3201	3202	3203	3204	3205	3206	3207	3208	3209	3210	3211	3212	3213	3214	3215
C90	3216	3217	3218	3219	3220	3221	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231
CA0	3232	3233	3234	3235	3236	3237	3238	3239	3240	3241	3242	3243	3244	3245	3246	3247
CB0	3248	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	3259	3260	3261	3262	3263
CC0	3264	3265	3266	3267	3268	3269	3270	3271	3272	3273	3274	3275	3276	3277	3278	3279
CD0	3280	3281	3282	3283	3284	3285	3286	3287	3288	3289	3290	3291	3292	3293	3294	3295
CE0	3296	3297	3298	3299	3300	3301	3302	3303	3304	3305	3306	3307	3308	3309	3310	3311
CF0	3312	3313	3314	3315	3316	3317	3318	3319	3320	3321	3322	3323	3324	3325	3326	3327
D00	3328	3329	3330	3331	3332	3333	3334	3335	3336	3337	3338	3339	3340	3341	3342	3343
D10	3344	3345	3346	3347	3348	3349	3350	3351	3352	3353	3354	3355	3356	3357	3358	3359
D20	3360	3361	3362	3363	3364	3365	3366	3367	3368	3369	3370	3371	3372	3373	3374	3375
D30	3376	3377	3378	3379	3380	3381	3382	3383	3384	3385	3386	3387	3388	3389	3390	3391
D40	3392	3393	3394	3395	3396	3397	3398	3399	3400	3401	3402	3403	3404	3405	3406	3407
D50	3408	3409	3410	3411	3412	3413	3414	3415	3416	3417	3418	3419	3420	3421	3422	3423
D60	3424	3425	3426	3427	3428	3429	3430	3431	3432	3433	3434	3435	3436	3437	3438	3439
D70	3440	3441	3442	3443	3444	3445	3446	3447	3448	3449	3450	3451	3452	3453	3454	3455

HEXADECIMAL-DECIMAL CONVERSION TABLE (Cont)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
D80	3456	3457	3458	3459	3460	3461	3462	3463	3464	3465	3466	3467	3468	3469	3470	3471
D90	3472	3473	3474	3475	3476	3477	3478	3479	3480	3481	3482	3483	3484	3485	3486	3487
DA0	3488	3489	3490	3491	3492	3493	3494	3495	3496	3497	3498	3499	3500	3501	3502	3503
DB0	3504	3505	3506	3507	3508	3509	3510	3511	3512	3513	3514	3515	3516	3517	3518	3519
DC0	3520	3521	3522	3523	3524	3525	3526	3527	3528	3529	3530	3531	3532	3533	3534	3535
DD0	3536	3537	3538	3539	3540	3541	3542	3543	3544	3545	3546	3547	3548	3549	3550	3551
DE0	3552	3553	3554	3555	3556	3557	3558	3559	3560	3561	3562	3563	3564	3565	3566	3567
DF0	3568	3569	3570	3571	3572	3573	3574	3575	3576	3577	3578	3579	3580	3581	3582	3583
E00	3584	3585	3586	3587	3588	3589	3590	3591	3592	3593	3594	3595	3596	3597	3598	3599
E10	3600	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610	3611	3612	3613	3614	3615
E20	3616	3617	3618	3619	3620	3621	3622	3623	3624	3625	3626	3627	3628	3629	3630	3631
E30	3632	3633	3634	3635	3636	3637	3638	3639	3640	3641	3642	3643	3644	3645	3646	3647
E40	3648	3649	3650	3651	3652	3653	3654	3655	3656	3657	3658	3659	3660	3661	3662	3663
E50	3664	3665	3666	3667	3668	3669	3670	3671	3672	3673	3674	3675	3676	3677	3678	3679
E60	3680	3681	3682	3683	3684	3685	3686	3687	3688	3689	3690	3691	3692	3693	3694	3695
E70	3696	3697	3698	3699	3700	3701	3702	3703	3704	3705	3706	3707	3708	3709	3710	3711
E80	3712	3713	3714	3715	3716	3717	3718	3719	3720	3721	3722	3723	3724	3725	3726	3727
E90	3728	3729	3730	3731	3731	3733	3734	3735	3736	3737	3738	3739	3740	3741	3742	3743
EA0	3744	3745	3746	3747	3748	3749	3750	3751	3752	3753	3754	3755	3756	3757	3758	3759
EB0	3760	3761	3762	3763	3764	3765	3766	3767	3768	3769	3770	3771	3772	3773	3774	3775
EC0	3776	3777	3778	3779	3780	3781	3782	3783	3784	3785	3786	3787	3788	3789	3790	3791
ED0	3792	3793	3794	3795	3796	3797	3798	3799	3800	3801	3802	3803	3804	3805	3806	3807
EE0	3808	3809	3810	3811	3812	3813	3814	3815	3816	3817	3818	3819	3820	3821	3822	3823
EF0	3824	3825	3826	3827	3827	3829	3830	3831	3832	3833	3834	3835	3836	3837	3838	3839

HEXADECIMAL-DECIMAL CONVERSION TABLE (cont)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
F00	3840	3841	3842	3843	3844	3845	3846	3847	3848	3849	3850	3851	3852	3853	3854	3855
F10	3856	3857	3858	3859	3860	3861	3862	3863	3864	3865	3866	3867	3868	3869	3870	3871
F20	3872	3873	3874	3875	3876	3877	3878	3879	3880	3881	3882	3883	3884	3885	3886	3887
F30	3888	3889	3890	3891	3892	3893	3894	3895	3896	3897	3898	3899	3900	3901	3902	3902
F40	3904	3905	3906	3907	3908	3909	3910	3911	3912	3913	3914	3915	3916	3917	3918	3919
F50	3920	3921	3922	3923	3924	3925	3926	3927	3928	3929	3930	3931	3932	3933	3934	3935
F60	3936	3937	3938	3939	3940	3941	3942	3943	3944	3945	3946	3947	3948	3949	3950	3951
F70	3952	3953	3954	3955	3956	3957	3958	3959	3960	3961	3962	3963	3964	3965	3966	3967
F80	3968	3969	3970	3971	3972	3973	3974	3975	3976	3977	3978	3979	3980	3981	3982	3983
F90	3984	3985	3986	3987	3988	3989	3990	3991	3992	3993	3994	3995	3996	3997	3998	3999
FA0	4000	4001	4002	4003	4004	4005	4006	4007	4008	4009	4010	4011	4012	4013	4014	4015
FB0	4016	4017	4018	4019	4020	4021	4022	4023	4024	4025	4026	4027	4028	4029	4030	4031
FC0	4032	4033	4034	4035	4036	4037	4038	4039	4040	4041	4042	4043	4044	4045	4046	4047
FD0	4048	4049	4050	4051	4052	4053	4054	4055	4056	4057	4058	4059	4060	4061	4062	4063
FE0	4064	4065	4066	4067	4068	4069	4070	4071	4072	4073	4074	4075	4076	4077	4078	4079
FF0	4080	4081	4082	4083	4084	4085	4086	4087	4088	4089	4090	4091	4092	4093	4094	4095

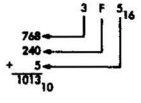
NUMBERING SYSTEMS

Decimal-Hexadecimal Conversion Table

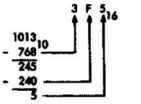
(DECIMAL = H x 16^{I})

H ↓ I → 6	5	4	3	2	1	0
1	16777216	1048576	65536	4096	256	16	1
2	33554432	2097152	131072	8192	512	32	2
3	50331648	3145728	196608	12288	768	48	3
4	67108864	4194304	262144	16384	1024	64	4
5	83886080	5242880	327680	20480	1280	80	5
6	100663296	6291456	393216	24576	1536	96	6
7	117440512	7340032	458752	28672	1792	112	7
8	134217728	8388608	524288	32768	2048	128	8
9	150994944	9437184	589824	36864	2304	144	9
A	167772160	10485760	655360	40960	2560	160	10
B	184549376	11534336	720896	45056	2816	176	11
C	201326592	12582912	786432	49152	3072	192	12
D	218103808	13631488	851968	53248	3328	208	13
E	234881024	14680064	917504	57344	3584	224	14
F	251658240	15728640	983040	61440	3840	240	15

HEXADECIMAL TO DECIMAL



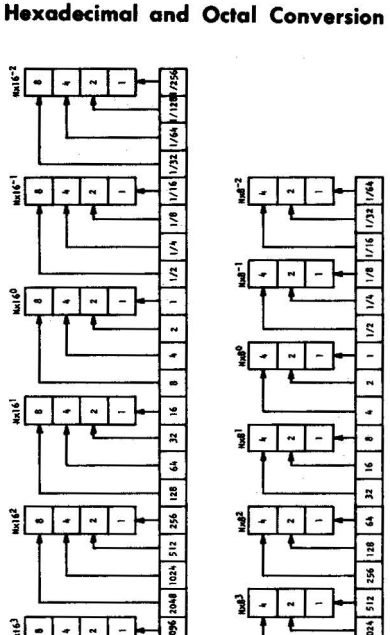
DECIMAL TO HEXADECIMAL



Hexadecimal to Decimal. Find the decimal value for each hexadecimal digit according to its position. Add these to obtain the decimal equivalent.

Decimal to Hexadecimal. Find the next lower decimal number and its Hexadecimal equivalent. Subtract and use difference to find the next decimal value and hexadecimal equivalent until the complete number is developed.

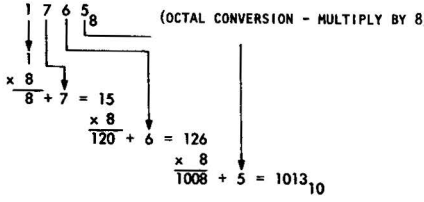
Binary to Hexadecimal and Octal Conversion Table

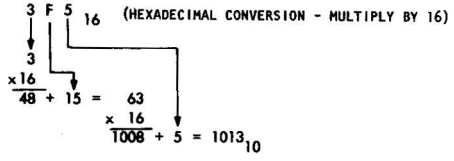


NUMBERING SYSTEMS

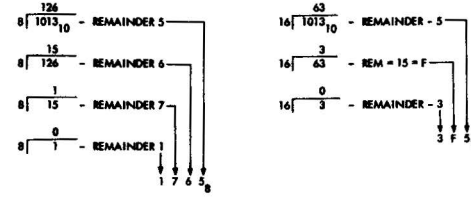
SAMPLE CONVERSION PROBLEMS

Octal (1765_8) and Hexadecimal ($3F5_{16}$) to Decimal (1013_{10})

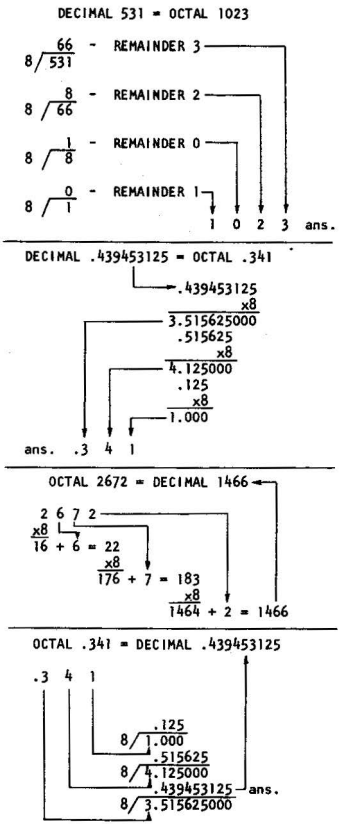




Decimal (1013_{10}) to Hexadecimal ($3F5_{16}$) and Octal (1765_8)



Decimal and Octal Sample Conversion Problems



NUMBERING SYSTEMS

Hexadecimal Addition Table

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

PRINTED IN U.S. AMERICA 5000276

COLLATING INFORMATION

All characters are collated according to their internal binary value. Because the B 6500/B 6700 has the capability of representing characters internally in either BCL or EBCDIC, and because characters are collated according to their internal representation (not necessarily the same as their external mode) a variety of collating sequences is possible. The following table may be used to determine the applicable collating sequence.

Input Mode	Output Mode	Internal Mode	Collating Sequence
BCL	BCL	BCL	BCL (BCL internal)
BCL	EBCDIC	EBCDIC	BCL Translated to EBCDIC
BCL	BCL	EBCDIC	BCL Translated to EBCDIC
EBCDIC	EBCDIC	EBCDIC	EBCDIC
EBCDIC	BCL	EBCDIC	BCL Translated to EBCDIC

Character Representation

The BCL, EBCDIC, and USASCII graphics are the same except as follows:

BCL	024/026 CRD/CDS	EBCDIC (029)	USASCII
[	& 8 4	<	
]	0 8 6	>	
<	& 8 6	+	
>	8 6	=	
≤	- 8 7	NOT (¬)	^
≥	8 7	"	'
=	0 8 5	—	
≠	0 8 2	N/G (0-8-2)	—
←	& 8 7	|	\
"	0 8 7	?	
:	8 5	'	
;	- 8 6	;	
X	- 0	MZ (N/K)	}
÷	& 0	PZ (N/K)	{
(	& 8 5	(	
)	- 8 5	)	
? *	8 2	:	
? *	& 8 2	[(N/K)	
? *	- 8 2	] (N/K)	

* Invalid Character

N/G-no graphic

N/K-not on keyboard

DRUM CARD

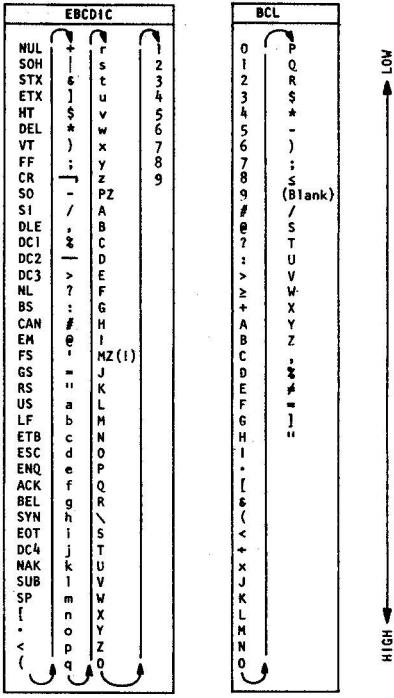
PROG 1	PROG 2	Operation
&	4	Field Definition
-	5	Start Auto Skip
0	6	Start Auto Dupl.
1	7	Alpha Shift

COLLATING

A BCL plus sign is never translated to an EBCDIC PZ (plus zero) sign, although the EBCDIC PZ is translated to a BCL plus sign.

EBCDIC 1110 0000 is translated to BCL 00 0000 with an additional flag bit on the next to most significant bit line (7th bit). As the print drums have 64 graphics and space this signal can be used to print the 64th graphic. The 64th graphic is a "CR" for BCL drums and a "¢" for EBCDIC drums.

Collating Sequences



EBCDIC COLLATING SEQUENCE

EBCDIC Character	Hex. Code	Internal Code	Card Code Zone Number	
NUL	00	0000 0000	12-0-9-	8-1
SOH	01	0000 0001	12-9-	1
STX	02	0000 0010	12-9-	2
ETX	03	0000 0011	12-9-	3
HT	05	0000 0101	12-9-	5
DEL	07	0000 0111	12-9-	7
VT	0B	0000 1011	12-9-	8-3
FF	0C	0000 1100	12-9-	8-4
CR	0D	0000 1101	12-9-	8-5
SO	0E	0000 1110	12-9-	8-6
SI	0F	0000 1111	12-9-	8-7
DLE	10	0001 0000	12-11-9-	8-1
DC1	11	0001 0001	11-9-	1
DC2	12	0001 0010	11-9-	2
DC3	13	0001 0011	11-9-	3
NL	15	0001 0101	11-9	5
BS	16	0001 0110	11-9-	6
CAN	18	0001 1000	11-9-	8
EM	19	0001 1001	11-9-	8-1
FS	1C	0001 1100	11-9-	8-4
GS	1D	0001 1101	11-9-	8-5
RS	1E	0001 1110	11-9-	8-6
US	1F	0001 1111	11-9-	8-7
LF	25	0010 0101	0-9-	5
ETB	26	0010 0110	0-9-	6
ESC	27	0010 0110	0-9-	7
ENQ	2D	0010 1101	0-9-	8-5
ACK	2E	0010 1110	0-9-	8-6
BEL	2F	0010 1111	0-9-	8-7
SYN	32	0011 0010	9-	2
EOT	37	0011 0111	9-	7
DC4	3C	0011 1100	9-	8-4
NAK	3D	0011 1101	9-	8-5
SUB	3F	0011 1111	9-	8-7
SP	40	0100 0000	(No Punches)	
[	4A	0100 1010	12-	8-2
.	4B	0100 1011	12-	8-3
<	4C	0100 1100	12-	8-4
(	4D	0100 1101	12-	8-5
+	4E	0100 1110	12-	8-6
| (↵)	4F	0100 1111	12-	8-7

LOW ↑ ↓ HIGH

COLLATING

EBCDIC COLLATING SEQUENCE (Cont)

EBCDIC Character	Hex. Code	Internal Code	Card Code Zone	Number
&	50	0101 0000	12-	-
]	5A	0101 1010	11-	8-2
$	5B	0101 1011	11-	8-3
*	5C	0101 1100	11-	8-4
)	5D	0101 1101	11-	8-5
;	5E	0101 1110	11-	8-6
¬ (≤)	5F	0101 1111	11-	8-7
- (Dash)	60	0110 0000	11-	-
/	61	0110 0001	0-	1
, (Comma)	6B	0110 1011	0-	8-3
%	6C	0110 1100	0-	8-4
_ (≠)	6D	0110 1101	0-	8-5
>	6E	0110 1110	0-	8-6
?	6F	0110 1111	0-	8-7
:	7A	0111 1010	-	8-2
#	7B	0111 1011	-	8-3
@	7C	0111 1100	-	8-4
' (≥)	7D	0111 1101	-	8-5
=	7E	0111 1110	-	8-6
"	7F	0111 1111	-	8-7
a	81	1000 0001	12-0-	1
b	82	1000 0010	12-0-	2
c	83	1000 0011	12-0-	3
d	84	1000 0100	12-0-	4
e	85	1000 0101	12-0-	5
f	86	1000 0110	12-0-	6
g	87	1000 0111	12-0-	7
h	88	1000 1000	12-0-	8
i	89	1000 1001	12-0-	9
j	91	1001 0001	12-11-	1
k	92	1001 0010	12-11-	2
l	93	1001 0011	12-11-	3
m	94	1001 0100	12-11-	4
n	95	1001 0101	12-11-	5
o	96	1001 0110	12-11-	6
p	97	1001 0111	12-11-	7
q	98	1001 1000	12-11-	8
r	99	1001 1001	12-11-	9

LOW ▲ HIGH ▼

EBCDIC COLLATING SEQUENCE (Cont)

EBCDIC Character	Hex. Code	Internal Code	Card Code Zone Number	
s	A2	1010 0010	11-0-	2
t	A3	1010 0011	11-0-	3
u	A4	1010 0100	11-0-	4
v	A5	1010 0101	11-0-	5
w	A6	1010 0110	11-0-	6
x	A7	1010 0111	11-0-	7
y	A8	1010 1000	11-0-	8
z	A9	1010 1001	11-0-	9
PZ (+)	C0	1100 0000	12-0	
A	C1	1100 0001	12-	1
B	C2	1100 0010	12-	2
C	C3	1100 0011	12-	3
D	C4	1100 0100	12-	4
E	C5	1100 0101	12-	5
F	C6	1100 0110	12-	6
G	C7	1100 0111	12-	7
H	C8	1100 1000	12-	8
I	C9	1100 1001	12-	9
MZ (!)	D0	1101 0000	11-	0
J	D1	1101 0001	11-	1
K	D2	1101 0010	11-	2
L	D3	1101 0011	11-	3
M	D4	1101 0100	11-	4
N	D5	1101 0101	11-	5
O	D6	1101 0110	11-	6
P	D7	1101 0111	11-	7
Q	D8	1101 1000	11-	8
R	D9	1101 1001	11-	9
\ (CR) (¢)	E0	1110 0000	0-	8-2
S	E2	1110 0010	0-	2
T	E3	1110 0011	0-	3
U	E4	1110 0100	0-	4
V	E5	1110 0101	0-	5
W	E6	1110 0110	0-	6
X	E7	1110 0111	0-	7
Y	E8	1110 1000	0-	8
Z	E9	1110 1001	0-	9

LOW ▲ ▼ HIGH

COLLATING

EBCDIC COLLATING SEQUENCE (Cont)

EBCDIC Character	Hex. Code	Internal Code	Card Code Zone	Number
0	F0	1111 0000	-	0
1	F1	1111 0001	-	1
2	F2	1111 0010	-	2
3	F3	1111 0011	-	3
4	F4	1111 0100	-	4
5	F5	1111 0101	-	5
6	F6	1111 0110	-	6
7	F7	1111 0111	-	7
8	F8	1111 1000	-	8
9	F9	1111 1001	-	9

LOW ▲ HIGH ▼

BCL COLLATING SEQUENCE (BCL INTERNAL)

BCL Character	BCL Octal	BCL Hex	BCL Internal BA 8421	BCL External BA 8421	Card Code Zone	Number
0	00	00	00 0000	00 1010	-	0
1	01	01	00 0001	00 0001	-	1
2	02	02	00 0010	00 0010	-	2
3	03	03	00 0011	00 0011	-	3
4	04	04	00 0100	00 0100	-	4
5	05	05	00 0101	00 0101	-	5
6	06	06	00 0110	00 0110	-	6
7	07	07	00 0111	00 0111	-	7
8	10	08	00 1000	00 1000	-	8
9	11	09	00 1001	00 1001	-	9
#	12	0A	00 1010	00 1011	-	8-3
@	13	0B	00 1010	00 1100	-	8-4
?	14	0C	00 1100	00 0000	All other card codes	
:	15	0D	00 1101	00 1101	-	8-5
>	16	0E	00 1110	00 1110	-	8-6
≥	17	0F	00 1111	00 1111	-	8-7
+	20	10	01 0000	11 1010	12	0
A	21	11	01 0001	11 0001	12	1
B	22	12	01 0010	11 0010	12	2
C	23	13	01 0011	11 0011	12	3
D	24	14	01 0100	11 0100	12	4
E	25	15	01 0101	11 0101	12	5
F	26	16	01 0110	11 0110	12	6
G	27	17	01 0111	11 0111	12	7

LOW ▲ HIGH ▼

BCL COLLATING SEQUENCE (BCL INTERNAL) (Cont)

BCL Character	BCL Octal	BCL Hex	BCL Internal BA 8421	BCL External BA 8421	Card Code Zone	Card Code Number
H	30	18	01 1000	11 1000	12	8
I	31	19	01 1001	11 1001	12	9
.	32	1A	01 1010	11 1011	12	8-3
[	33	1B	01 1011	11 1100	12	8-4
&	34	1C	01 1100	11 0000	12	-
(	35	1D	01 1101	11 1101	12	8-5
<	36	1E	01 1110	11 1110	12	8-6
+	37	1F	01 1111	11 1111	12	8-7
x(Mult.)	40	20	10 0000	10 1010	11	0
J	41	21	10 0001	10 0001	11	1
K	42	22	10 0010	10 0010	11	2
L	43	23	10 0011	10 0011	11	3
M	44	24	10 0100	10 0100	11	4
N	45	25	10 0101	10 0101	11	5
O	46	26	10 0110	10 0110	11	6
P	47	27	10 0111	10 0111	11	7
Q	50	28	10 1000	10 1000	11	8
R	51	29	10 1001	10 1001	11	9
$	52	2A	10 1010	10 1011	11	8-3
*	53	2B	10 1011	10 1100	11	8-4
-	54	2C	10 1100	10 0000	11	-
)	55	2D	10 1101	10 1101	11	8-5
;	56	2E	10 1110	10 1110	11	8-6
≤	57	2F	10 1111	10 1111	11	8-7
Blank	60	30	11 0000	01 0000	-	-
/	61	31	11 0001	01 0001	0	1
S	62	32	11 0010	01 0010	0	2
T	63	33	11 0011	01 0011	0	3
U	64	34	11 0100	01 0100	0	4
V	65	35	11 0101	01 0101	0	5
W	66	36	11 0110	01 0110	0	6
X	67	37	11 0111	01 0111	0	7
Y	70	38	11 1000	01 1000	0	8
Z	71	39	11 1001	01 1001	0	9
,	72	3A	11 1010	01 1011	0	8-3
%	73	3B	11 1011	01 1100	0	8-4
≠	74	3C	11 1100	01 1010	0	8-2
=	75	3D	11 1101	01 1101	0	8-5
]	76	3E	11 1110	01 1110	0	8-6
"	77	3F	11 1111	01 1111	0	8-7

LOW ↑ ↓ HIGH

COLLATING

COLLATING SEQUENCE - BCL TRANSLATED TO EBCDIC

BCL Character	BCL External BA 4321	BCL Hex.	BCL Octal	Translated EBCDIC Code	EBCDIC Hex	Card Code Zone	Card Code Number
(Blank)	01 0000	10	20	0100 0000	40	-	-
[	11 1100	3C	74	0100 1010	4A	12	8-4
.	11 1011	3B	73	0100 1011	4B	12	8-3
<	11 1110	3E	76	0100 1100	4C	12	8-6
(	11 1101	3D	75	0100 1101	4D	12	8-5
+	11 1010	3A	72	0100 1110	4E	12	0
+	11 1111	3F	77	0100 1111	4F	12	8-7
&	11 0000	30	60	0101 0000	50	12	-
]	01 1110	1E	36	0101 1010	5A	0	8-6
$	10 1011	2B	53	0101 1011	5B	11	8-3
*	10 1100	2C	54	0101 1100	5C	11	8-4
)	10 1101	2D	55	0101 1101	5D	11	8-5
;	10 1110	2E	56	0101 1110	5E	11	8-6
≤	10 1111	2F	57	0101 1111	5F	11	8-7
-	10 0000	20	40	0110 0000	60	11	-
/	01 0001	11	21	0110 0001	61	0	1
,	01 1011	1B	33	0110 1011	6B	0	8-3
%	01 1100	1C	34	0110 1100	6C	0	8-4
≠	01 1010	1A	32	0110 1101	6D	0	8-2
>	00 1110	0E	16	0110 1110	6E	-	8-6
?	00 0000	00	00	0110 1111	6F	All other card codes	
:	00 1101	0D	15	0111 1010	7A	-	8-5
#	00 1011	0B	13	0111 1011	7B	-	8-3
@	00 1100	0C	14	0111 1100	7C	-	8-4
≥	00 1111	0F	17	0111 1101	7D	-	8-7
=	01 1101	1D	35	0111 1110	7E	0	8-5
"	01 1111	1F	37	0111 1111	7F	0	8-7
A	11 0001	31	61	1100 0001	C1	12	1
B	11 0010	32	62	1100 0010	C2	12	2
C	11 0011	33	63	1100 0011	C3	12	3
D	11 0100	34	64	1100 0100	C4	12	4
E	11 0101	35	65	1100 0101	C5	12	5
F	11 0110	36	66	1100 0110	C6	12	6
G	11 0111	37	67	1100 0111	C7	12	7
H	11 1000	38	70	1100 1000	C8	12	8
I	11 1001	39	71	1100 1001	C9	12	9

LOW ▲ ▼ HIGH

COLLATING SEQUENCE - BCL TRANSLATED TO EBCDIC (Cont)

BCL Character	BCL External BA 4321	BCL Hex.	BCL Octal	Translated EBCDIC Code	EBCDIC Hex.	Card Code Zone	Card Code Number
x(mult)	10 1010	2A	52	1101 0000	D0	11	0
J	10 0001	21	41	1101 0001	D1	11	1
K	10 0010	22	42	1101 0010	D2	11	2
L	10 0011	23	43	1101 0011	D3	11	3
M	10 0100	24	44	1101 0100	D4	11	4
N	10 0101	25	45	1101 0101	D5	11	5
O	10 0110	26	46	1101 0110	D6	11	6
P	10 0111	27	47	1101 0111	D7	11	7
Q	10 1000	28	50	1101 1000	D8	11	8
R	10 1001	29	51	1101 1001	D9	11	9
S	01 0010	12	22	1110 0010	E2	0	2
T	01 0011	13	23	1110 0011	E3	0	3
U	01 0100	14	24	1110 0100	E4	0	4
V	01 0101	15	25	1110 0101	E5	0	5
W	01 0110	16	26	1110 0110	E6	0	6
X	01 0111	17	27	1110 0111	E7	0	7
Y	01 1000	18	30	1110 1000	E8	0	8
Z	01 1001	19	31	1110 1001	E9	0	9
0	00 1010	0A	12	1111 0000	F0	-	0
1	00 0001	01	01	1111 0001	F1	-	1
2	00 0010	02	02	1111 0010	F2	-	2
3	00 0011	03	03	1111 0011	F3	-	3
4	00 0100	04	04	1111 0100	F4	-	4
5	00 0101	05	05	1111 0101	F5	-	5
6	00 0110	06	06	1111 0110	F6	-	6
7	00 0111	07	07	1111 0111	F7	-	7
8	00 1000	08	10	1111 1000	F8	-	8
9	00 1001	09	11	1111 1001	F9	-	9

LOW ▲ ▼ HIGH

COLLATING

XALGOL COLLATING SEQUENCE (B 5700 BCL)

BCL Character	BCL Octal	BCL Hex	BCL Internal BA 8421	BCL External BA 8421	Card Code Zone	Card Code Number
Blank	60	30	11 0000	01 0000	-	-
.	32	1A	01 1010	11 1011	12	8-3
[	33	1B	01 1011	11 1100	12	8-4
(	35	1D	01 1101	11 1101	12	8-5
<	36	1E	01 1110	11 1110	12	8-6
+	37	1F	01 1111	11 1111	12	8-7
&	34	1C	01 1100	11 0000	12	-
$	52	2A	10 1010	10 1011	11	8-3
*	53	2B	10 1011	10 1100	11	8-4
)	55	2D	10 1101	10 1101	11	8-5
;	56	2E	10 1110	10 1110	11	8-6
≤	57	2F	10 1111	10 1111	11	8-7
-	54	2C	10 1100	10 0000	11	-
/	61	31	11 0001	01 0001	0	1
,	72	3A	11 1010	01 1011	0	8-3
%	73	3B	11 1011	01 1100	0	8-4
=	75	3D	11 1101	01 1101	0	8-5
]	76	3E	11 1110	01 1110	0	8-6
"	77	3F	11 1111	01 1111	0	8-7
#	12	0A	00 1010	00 1011	-	8-3
@	13	0B	00 1011	00 1100	-	8-4
:	15	0D	00 1101	00 1101	-	8-5
>	16	0E	00 1110	00 1110	-	8-6
≥	17	0F	00 1111	00 1111	-	8-7
+	20	10	01 0000	11 1010	12	0
A	21	11	01 0001	11 0001	12	1
B	22	12	01 0010	11 0010	12	2
C	23	13	01 0011	11 0011	12	3
D	24	14	01 0100	11 0100	12	4
E	25	15	01 0101	11 0101	12	5
F	26	16	01 0110	11 0110	12	6
G	27	17	01 0111	11 0111	12	7
H	30	18	01 1000	11 1000	12	8
I	31	19	01 1001	11 1001	12	9
x	40	20	10 0000	10 1010	11	0
J	41	21	10 0001	10 0001	11	1
K	42	22	10 0010	10 0010	11	2
L	43	23	10 0011	10 0011	11	3
M	44	24	10 0100	10 0100	11	4
N	45	25	10 0101	10 0101	11	5

LOW ↑ COLLATING SEQUENCE ↓ HIGH

XALGOL COLLATING SEQUENCE (Cont)
(B 5700 BCL)

BCL Character	BCL Octal	BCL Hex	BCL Internal BA 8421	BCL External BA 8421	Card Code Zone	Card Code Number	
0	46	26	10 0110	10 0110	11	6	
P	47	27	10 0111	10 0111	11	7	LOW
Q	50	28	10 1000	10 1000	11	8	
R	51	29	10 1001	10 1001	11	9	
≠	74	3C	11 1100	01 1010	0	8-2	
S	62	32	11 0010	01 0010	0	2	
T	63	33	11 0011	01 0011	0	3	
U	64	34	11 0100	01 0100	0	4	
V	65	35	11 0101	01 0101	0	5	
W	66	36	11 0110	01 0110	0	6	
X	67	37	11 0111	01 0111	0	7	
Y	70	38	11 1000	01 1000	0	8	
Z	71	39	11 1001	01 1001	0	9	
0	00	00	00 0000	00 1010	-	0	
1	01	01	00 0001	00 0001	-	1	HIGH
2	02	02	00 0010	00 0010	-	2	
3	03	03	00 0011	00 0011	-	3	
4	04	04	00 0100	00 0100	-	4	
5	05	05	00 0101	00 0101	-	5	
6	06	06	00 0110	00 0110	-	6	
7	07	07	00 0111	00 0111	-	7	
8	10	08	00 1000	00 1000	-	8	
9	11	09	00 1001	00 1001	-	9	
?	14	0C	00 1100	00 0000	ALL OTHER CARD CODES		

COLLATING

FORTRAN BCD COLLATING SEQUENCE

BCD Character	Internal Representation Hex	Internal Representation Binary	Translation Binary	Translation Hex	Card Code Zone	Card Code Number	
. (period)	1A	01 1010	0100 1011	4B	12	8-3	Low
)	1B	01 1011	0100 1100	4C	12	8-4	
+	1C	01 1100	0101 0000	50	12		
$	2A	10 1010	0101 1011	5B	11	8-3	
*	2B	10 1011	0101 1100	5C	11	8-4	
;	2E	10 1110	0101 1110	5E	11	8-6	
¬	2F	10 1111	0101 1111	5F	11	8-7	
- (minus)	2C	10 1100	0110 0000	60	11		
/	31	11 0001	0110 0001	61	0	1	
, (comma)	3A	11 1010	0110 1011	6B	0	8-3	
(	3B	11 1011	0110 1100	6C	0	8-4	
≠	3D	11 1101	0110 1101	6D	0	8-5	
>	3E	11 1110	0110 1110	6E	0	8-6	
?	3F	11 1111	0110 1111	6F	0	8-7	
≡	0A	00 1010	0111 1011	7B		8-3	
@	0B	00 1011	0111 1100	7C		8-4	
∨	0D	00 1101	0111 1101	7D		8-5	
≠	0E	00 1110	0111 1110	7E		8-6	
∧	0F	00 1111	0111 1111	7E		8-7	
A	11	01 0001	1100 0001	C1	12	1	
B	12	01 0010	1100 0010	C2	12	2	
C	13	01 0011	1100 0011	C3	12	3	
D	14	01 0100	1100 0100	C4	12	4	
E	15	01 0101	1100 0101	C5	12	5	
F	16	01 0110	1100 0110	C6	12	6	
G	17	01 0111	1100 0111	C7	12	7	
H	18	01 1000	1100 1000	C8	12	8	
I	19	01 1001	1100 1001	C9	12	9	
J	21	10 0001	1101 0001	D1	11	1	
K	22	10 0010	1101 0010	D2	11	2	
L	23	10 0011	1101 0011	D3	11	3	
M	24	10 0100	1101 0100	D4	11	4	
N	25	10 0101	1100 0101	D5	11	5	
O	26	10 0110	1101 0110	D6	11	6	
P	27	10 0111	1101 0111	D7	11	7	
Q	28	10 1000	1101 1000	D8	11	8	
R	29	10 1001	1101 1001	D9	11	9	
S	31	11 0010	1110 0010	E2	0	2	
T	33	11 0011	1110 0011	E3	0	3	
U	34	11 0100	1110 0100	E4	0	4	
V	35	11 0101	1110 0101	E5	0	5	
W	36	11 0110	1110 0110	E6	0	6	
X	37	11 0111	1110 0111	E7	0	7	
Y	38	11 1000	1110 1000	E8	0	8	
Z	39	11 1001	1110 1001	E9	0	9	
0	00	00 0000	1111 0000	F0		0	
1	01	00 0001	1111 0001	F1		1	
2	02	00 0010	1111 0010	F2		2	High

FORTRAN BCD COLLATING SEQUENCE (Cont)

BCD Character	Internal Representation Hex	Binary	Internal Translation Binary	Hex	Card Code Zone Number
3	03	00 0011	1111 0011	F3	3
4	04	00 0100	1111 0100	F4	4
5	05	05 0101	1111 0101	F5	5
6	06	00 0110	1111 0110	F6	6
7	07	00 0111	1111 0111	F7	7
8	08	00 1000	1111 1000	F8	8
9	09	00 1001	1111 1001	F9	9

Low ↑ ↓ High

EXTENDED BINARY CODED DECIMAL INTERCHANGE CODES (EBCDIC)

b0 b7 b6 b5 / b4 b3 b2 b1 / COL / ROW	0	1	2	3	4	5	6	7	8	9	10 (A)	11 (B)	12 (C)	13 (D)	14 (E)	15 (F)
0	NUL 12-0-9-8-1	DLE 12-11-9-8-1	11-0-9-8-1	12-11-0-9-8-1	SP BLANK	& 12	– 11	12-11-0	12-0-9-1	12-11-9-1	11-0-9-1	12-11-0-9-1	FS(†) 12-0	MZ [‡] 11-0	\(CR)(‡) 0-9-8-2	0 0
1	SOH 12-9-1	DC1 11-9-1	0-9-1	9-1	12-0-9-1	12-11-9-1	/ 0-1	12-11-0-9-1	a 12-0-1	j 12-11-1	11-0-1	12-11-0-1	A 12-1	J 11-1	11-0-9-1	1 1
2	STX 12-9-2	DC2 11-9-2	0-9-2	SYN 9-2	12-0-9-2	12-11-9-2	11-0-9-2	12-11-0-9-2	b 12-0-2	k 12-11-2	s 11-0-2	12-11-0-2	B 12-2	K 11-2	S 0-2	2 2
3	ETX 12-9-3	DC3 11-9-3	0-9-3	9-3	12-0-9-3	12-11-9-3	11-0-9-3	12-11-0-9-3	c 12-0-3	l 12-11-3	t 11-0-3	12-11-0-3	C 12-3	L 11-3	T 0-3	3 3
4	12-9-4	11-9-4	0-9-4	9-4	12-0-9-4	12-11-9-4	11-0-9-4	12-11-0-9-4	d 12-0-4	m 12-11-4	u 11-0-4	12-11-0-4	D 12-4	M 11-4	U 0-4	4 4
5	HT 12-9-5	NL 11-9-5	LF 0-9-5	9-5	12-0-9-5	12-11-9-5	11-0-9-5	12-11-0-9-5	e 12-0-5	n 12-11-5	v 11-0-5	12-11-0-5	E 12-5	N 11-5	V 0-5	5 5
6	12-9-6	BS 11-9-6	ETB 0-9-6	9-6	12-0-9-6	12-11-9-6	11-0-9-6	12-11-0-9-6	f 12-0-6	o 12-11-6	w 11-0-6	12-11-0-6	F 12-6	O 11-6	W 0-6	6 6
7	DEL 12-9-7	11-9-7	ESC 0-9-7	EOT 9-7	12-0-9-7	12-11-9-7	11-0-9-7	12-11-0-9-7	g 12-0-7	p 12-11-7	x 11-0-7	12-11-0-7	G 12-7	P 11-7	X 0-7	7 7
8	12-9-8	CAN 11-9-8	0-9-8	9-8	12-0-9-8	— 12-11-9-8	11-0-9-8	12-11-0-9-8	h 12-0-8	q 12-11-8	y 11-0-8	12-11-0-8	H 12-8	Q 11-8	Y 0-8	8 8
9	12-9-8-1	EM 11-9-8-1	0-9-8-1	9-8-1	12-8-1	11-8-1	0-8-1	8-1	i 12-0-9	r 12-11-9	z 11-0-9	12-11-0-9	I 12-9	R 11-9	Z 0-9	9 9
10 (A)	12-9-8-2	11-9-8-2	0-9-8-2	9-8-2	[12-8-2	] 11-8-2	12-11	: 8-2	12-0-8-2	12-11-8-2	11-0-8-2	12-11-0-8-2	12-0-9-8-2	12-11-9-8-2	11-0-9-8-2	12-11-0-9-8-2
11 (B)	VT 12-9-8-3	11-9-8-3	0-9-8-3	9-8-3	$ 12-8-3	11-8-3	0-8-3	# 8-3	12-0-8-3	12-11-8-3	11-0-8-3	12-11-0-8-3	12-0-9-8-3	12-11-9-8-3	11-0-9-8-3	12-11-0-9-8-3
12 (C)	FF 12-9-8-4	FS 11-9-8-4	0-9-8-4	DC4 9-8-4	< 12-8-4	@ 11-8-4	% 0-8-4	@ 8-4	12-0-8-4	12-11-8-4	11-0-8-4	12-11-0-8-4	12-0-9-8-4	12-11-9-8-4	11-0-9-8-4	12-11-0-9-8-4
13 (D)	CR 12-9-8-5	GS 11-9-8-5	ENQ 0-9-8-5	NAK 9-8-5	(12-8-5	) 11-8-5	(U3)(≠) 0-8-5	= 8-5	12-0-8-5	12-11-8-5	11-0-8-5	12-11-0-8-5	12-0-9-8-5	12-11-9-8-5	11-0-9-8-5	12-11-0-9-8-5
14 (E)	SO 12-9-8-6	RS 11-9-8-6	ACK 0-9-8-6	9-8-6	+ 12-8-6	; 11-8-6	> 0-8-6	" 8-6	12-0-8-6	12-11-8-6	11-0-8-6	12-11-0-8-6	12-0-9-8-6	12-11-9-8-6	11-0-9-8-6	12-11-0-9-8-6
15 (F)	SI 12-9-8-7	US 11-9-8-7	BEL 0-9-8-7	SUB 9-8-7	|(¬) 12-8-7	¬(≤) 11-8-7	? 0-8-7	' 8-7	12-0-8-7	12-11-8-7	11-0-8-7	12-11-0-8-7	DELIMETER 12-0-9-8-7	12-11-9-8-7	11-0-9-8-7	12-11-0-9-8-7

* INTERNAL COLLATING SEQUENCE - 0 0 0 0/0 0 0 0 TO 1 1 1 1/1 1 1 1

DATA TRANSMISSION CODES

Baudot Code

b4	b3	b2	b1	Column / Row #	0	1	2	3
0	0	0	0	0	BLK	T	BLK	5
0	0	0	1	1	E	Z	3	"
0	0	1	0	2	LF	L	LF	3/4)
0	0	1	1	3	A	W	-	2
0	1	0	0	4	SPACE	H	SPACE	DIAMOND
0	1	0	1	5	S	Y	BELL	6
0	1	1	0	6	I	P	8	0
0	1	1	1	7	U	Q	7	1
1	0	0	0	8	CR	O	CR	9
1	0	0	1	9	D	B	$	5/8 ?
1	0	1	0	10(A)	R	G	4	&
1	0	1	1	11(B)	J	FIGS	'	FIGS
1	1	0	0	12(C)	N	M	7/8 .	.
1	1	0	1	13(D)	F	X	1/4 ,	/
1	1	1	0	14(E)	C	V	1/8 :	3/8 :
1	1	1	1	15(F)	K	LTRS	1/2 (	LTRS

PTTC/6 Code

b4	b3	b2	b1	Column / Row #	0	1	2	3	4	5	6	7
0	0	0	0	0	SPACE	@	-	+ &	SPACE	DELTA	BACK/	<
0	0	0	1	1	1	/	j	a	>	QUES	J	A
0	0	1	0	2	2	s	k	b	)	S	K	B
0	0	1	1	3	3	t	l	c	;	T	L	C
0	1	0	0	4	4	u	m	d	SBLANK	U	M	D
0	1	0	1	5	5	v	n	e	(	V	N	E
0	1	1	0	6	6	w	o	f	:	W	O	F
0	1	1	1	7	7	x	p	g	"	X	P	G
1	0	0	0	8	8	y	q	h	*	Y	Q	H
1	0	0	1	9	9	z	r	i	[	Z	R	I
1	0	1	0	10(A)	0	#	NU	PZ	]	GRPMRK	GAMMA	SQ. RT
1	0	1	1	11(B)	= +	,	$	.	SEGMARK	.	V.BAR	
1	1	0	0	12(C)	PN	BY	RFS	PF	PN	BY	RES	PF
1	1	0	1	13(D)	RS	LF	NL	HT	BS	LF	NL	HT
1	1	1	0	14(E)	UC	EOR	BS	LC	UC	EOB	BS	LC
1	1	1	1	15(F)	EOT	PRE	IL	DEL	EOT	PRF	IL	DEL

DATA TRANSMISSION CODES

USASCII X3.4-1963 Code

Bits b_7 b_6 b_5 / b_4 b_3 b_2 b_1 (Row / Column)	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0 0 0	0	1	2	3	4	5	6	7
0 0 0 1	1	SOM	DC_1	!	1	A	Q	a	q
0 0 1 0	2	EOA	DC_2	"	2	B	R	b	r
0 0 1 1	3	EOM	DC_3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC_4	$	4	D	T	d	t
0 1 0 1	5	WRU	ERR	%	5	E	U	e	u
0 1 1 0	6	RU	SYNC	&	6	F	V	f	v
0 1 1 1	7	BELL	LEM	' (APOS)	7	G	W	g	w
1 0 0 0	8	FE_0	S0	(	8	H	X	h	x
1 0 0 1	9	HT/SK	S1	)	9	I	Y	i	y
1 0 1 0	10	LF	S2	*	:	J	Z	j	z
1 0 1 1	11	VT	S3	+	;	K	[	k	{
1 1 0 0	12	FF	S4	, (COMMA)	<	L	\	l	|
1 1 0 1	13	CR	S5	-	=	M	]	m	}
1 1 1 0	14	SO	S6	.	>	N	↑	n	~
1 1 1 1	15	SI	S7	/	?	O	←	o	DEL

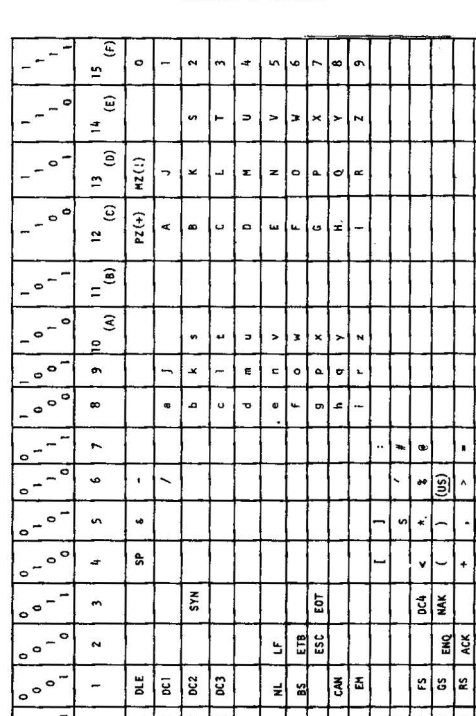
(1) Unassigned Control

UNUSED	UNUSED	UNUSED	UNUSED	UNUSED	UNUSED	UNUSED	UNUSED
ACK	(1)	ESC	DEL

USASCII X3.4-1967 Code

Bits b_7 b_6 b_5 / b_4 b_3 b_2 b_1 (Row / Column)	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	|
1 1 0 1	13	CR	GS	-	=	M	]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DEL

EBCDIC Code



B b8 / I b7 / T b6 / S b5		0 0 0 0	0 0 0 1	0 0 1 0	0 0 1 1	0 1 0 0	0 1 0 1	0 1 1 0	0 1 1 1	1 0 0 0	1 0 0 1	1 0 1 0	1 0 1 1	1 1 0 0	1 1 0 1	1 1 1 0	1 1 1 1
b4 b3 b2 b1	COL → / ROW ↓	0	1	2	3	4	5	6	7	8	9	10 (A)	11 (B)	12 (C)	13 (D)	14 (E)	15 (F)
0 0 0 0	0	NUL	DLE			SP	&	-						PZ(+)	MZ(-)		0
0 0 0 1	1	SOH	DC1					/		a	j			A	J		1
0 0 1 0	2	STX	DC2		SYN					b	k	s		B	K	S	2
0 0 1 1	3	ETX	DC3							c	l	t		C	L	T	3
0 1 0 0	4									d	m	u		D	M	U	4
0 1 0 1	5	HT	NL	LF						e	n	v		E	N	V	5
0 1 1 0	6		BS	ETB						f	o	w		F	O	W	6
0 1 1 1	7	DEL		ESC	EOT					g	p	x		G	P	X	7
1 0 0 0	8		CAN							h	q	y		H	Q	Y	8
1 0 0 1	9		EM							i	r	z		I	R	Z	9
1 0 1 0	10 (A)					[	]	:									
1 0 1 1	11 (B)	VT				.	$	#									
1 1 0 0	12 (C)	FF	FS		DC4	<	*	@									
1 1 0 1	13 (D)	CR	GS	ENQ	NAK	(	)	(US)									
1 1 1 0	14 (E)	SO	RS	ACK		+	;	=									
1 1 1 1	15 (F)	SI	US	BEL	SUB	|	¬	?	"					DELIMITER			

1963, 1967 USASCII Characters

COLUMN / ROW	CHARACTER: USASCII X3.4 - 1963 / USASCII X3.4 - 1967	CHARACTER: NAME/FUNCTION
0/0	NULL NUL	Null: The all zeros character which may serve to accomplish time fill and media fill.
0/1	SOM	Start of Message: It is used in conjunction with EOA, EOM, and EOT for messages on tapes where the message is to be sent automatically.
	SOH	Start of Heading: A communication control character used at the beginning of a sequence of characters which constitute a machine-sensible address or routing information. Such a sequence is referred to as the "Heading." An STX character has the effect of terminating a heading.
0/2	EOA	End of Address: This character, together with SOH, will be used to define the section of perforated tape in which the call-directing codes of the addressee are contained.
	STX	Start of Text: A communication control character which precedes a sequence of characters that are to be treated as an entity and entirely transmitted through to the ultimate destination. Such a sequence is referred to as "text." STX may be used to terminate a sequence of characters started by SOH.
0/3	EOM	End of Message: It may be used to separate individual messages which are sent in sequence on a single transmission between two stations (see SOM).
	ETX	End of Text: A communication control character used to terminate a sequence of characters started with STX and transmitted as an entity.
0/4	EOT	End of Transmission: A communication control character used to indicate the conclusion of a transmission which may have contained one or more texts and any associated headings.
	EOT	
0/5	WRU	Enquiry: A communication control character used in data communication systems as a request for a response from a remote station. It may be used as a "Who Are You" (WRU) to obtain identification, or may be used to obtain station status, or both.
	ENQ	

1963, 1967 USASCII Characters (Cont)

COLUMN / ROW	CHARACTER: USASCII X3.4 - 1963 / USASCII X3.4 - 1967	CHARACTER: NAME/FUNCTION
0/6	RU	Are You: Use of this character for confirmation type of answer back has been discontinued until a more suitable arrangement can be devised.
	ACK	Acknowledge: A communication control character transmitted by a receiver as an affirmative response to a sender.
0/7	BELL / BELL	Bell: A character for use when there is a need to call for human attention. It may control alarm or attention devices.
0/8	FE / BS	Backspace: A format effector that controls the movement of the printing mechanism one print position backward on the same print line.
0/9	HT/SK / HT	Horizontal Tabulation: A format effector that controls the movement of the printing mechanism to the next in a series of predetermined positions along the print line. (Applicable also to the skip function on punched cards.)
0/10	LF / LF	Line Feed: A format effector that controls the movement of the paper one line at a time.
0/11	VT / VT	Vertical Tabulation: A format effector that controls the movement of paper to the next in a series of predetermined print lines.
0/12	FF / FF	Form Feed: A format effector that controls the movement of the printing position to the first predetermined printing line on the next form or page.
0/13	CR / CR	Carriage Return: A format effector that controls the movement of the print mechanism to the first print position on the same print line.
0/14	SO / SO	Shift Out: A control character indicating that the code combinations that follow shall be interpreted as outside of the character set of the standard code table until a Shift In character is reached.

DATA TRANSMISSION CODES

1963, 1967 USASCII Characters (Cont)

COLUMN / ROW	CHARACTER: USASCII X3.4 - 1963 / USASCII X3.4 - 1967	CHARACTER: NAME/FUNCTION
0/15	SI SI	Shift In: A control character indicating that the code combinations that follow shall be interpreted according to the standard code table.
1/0	DC_0 DLE	Data Link Escape: A communication control character that will change the meaning of a limited number of contiguously following characters. It is used exclusively to provide supplementary controls in data communication networks.
1/1 1/2 1/3 1/4	DC_1 DC_2 DC_3 DC_4	Device Controls: DC_1 (X-ON) turns the tape reader ON and DC_3 (X-OFF) turns the tape reader OFF in Models 33 and 35. DC_2 and DC_4 can be used as PUNCH-ON and PUNCH-OFF controls.
	DC_1 DC_2 DC_3 DC_4	Device Controls: Characters for the control of ancillary devices associated with data processing or telecommunication systems, more especially switching devices ON or OFF. (If a single "stop" control is required to interrupt or turn off ancillary devices, DC_4 is the preferred assignment.
1/5	ERR NAK	Negative Acknowledge: A communication control character transmitted by a receiver as a negative response to the sender.
1/6	SYN SYN	Synchronous Idle: A communication control character used by a synchronous transmission system in the absence of any other character to provide a signal from which synchronism may be achieved or retained.
	LEM	Logical End of Media: Used to indicate the end of usable information, as in "End-of-Card".
1/7	ETB	End of Transmission Block: A communication control character used to indicate the end of a block of data for communication purposes. ETB is used for blocking data where the block structure is not necessarily related to the processing format.

1963, 1967 USASCII Characters (Cont)

COLUMN / ROW	CHARACTER: USASCII X3.4 - 1963 / USASCII X3.4 - 1967	CHARACTER: NAME/FUNCTION
1/8	S0	Information Separators.
	CAN	Cancel: A control character used to indicate that the data with which it is sent is in error or is to be disregarded.
1/9	S1	Information Separators.
	EM	End of Medium: A control character associated with the sent data which may be used to identify the physical end of the medium, or the end of the used, or wanted, portion of information recorded on a medium. (The position of this character does not necessarily correspond to the physical end of the medium).
1/10	S2	Information Separator.
	SUB	Substitute: A character that may be substituted for a character which is determined to be invalid or in error.
1/11	S3	Information Separator.
	ESC	Escape: A control character intended to provide code extension (supplementary characters) in general information interchange. The Escape character itself is a prefix affecting the interpretation of a limited number of contiguously following characters.
1/12 1/13	S4 S5 S6 S7	Information Separators.
1/14 1/15	FS GS RS US	File Separator, Group Separator, Record Separator, and Unit Separator: These information separators may be used within data in optional fashion, except that their hierarchical relationship shall be: FS is the most inclusive, then GS, then RS, and US is least inclusive. (The content and length of a File, Group, Record or Unit are not specified.)

DATA TRANSMISSION CODES

1963, 1967 USASCII Characters (Cont)

COLUMN / ROW	CHARACTER: USASCII X3.4 - 1963 / USASCII X3.4 - 1967	NAME/FUNCTION
2/0	SP / SP	Space: A normally non-printing graphic character used to separate words. It is also a format effector which controls the movement of the printing position, one printing position forward.
2/1	! / !	Exclamation Point.
2/2	" / "	Quotation Marks (Diaeresis).
2/3	# / #	Number Sign.
2/4	$ / $	Dollar Sign.
2/5	% / %	Percent.
2/6	& / &	Ampersand.
2/7	' / '	Apostrophe (Closing Single Quotation Mark; Acute Accent).
2/8	(/ (	Opening Parenthesis.
2/9	) /)	Closing Parenthesis.
2/10	* / *	Asterisk.
2/11	+ / +	Plus.

1963, 1967 USASCII Characters (Cont)

COLUMN / ROW	CHARACTER: USASCII X3.4 - 1963 / USASCII X3.4 - 1967	NAME/FUNCTION
2/12	, / ,	Comma (Cedilla).
2/13	- / -	Hyphen (Minus).
2/14	. / .	Period (Decimal Point).
2/15	/ /	Slant (Slash).
3/0	0 / 0	Figure Zero.
3/1	1 / 1	Figure One.
3/2	2 / 2	Figure Two.
3/3	3 / 3	Figure Three.
3/4	4 / 4	Figure Four.
3/5	5 / 5	Figure Five.
3/6	6 / 6	Figure Six.
3/7	7 / 7	Figure Seven.

DATA TRANSMISSION CODES

1963, 1967 USASCII Characters (Cont)

COLUMN / ROW	CHARACTER	
	USASCII X3.4 - 1963 USASCII X3.4 - 1967	NAME/FUNCTION
3/8	8 8	Figure Eight.
3/9	9 9	Figure Nine.
3/10	: :	Colon.
3/11	; ;	Semicolon.
3/12	< <	Less Than.
3/13	= =	Equals.
3/14	> >	Greater Than.
3/15	? ?	Question Mark.
4/0	@ @	Commercial At.
4/1	A A	Upper Case Letter A.
4/2	B B	Upper Case Letter B.
4/3	C C	Upper Case Letter C.

1963, 1967 USASCII Characters (Cont)

COLUMN / ROW	CHARACTER	
	USASCII X3.4 - 1963 USASCII X3.4 - 1967	NAME/FUNCTION
4/4	D D	Upper Case Letter D.
4/5	E E	Upper Case Letter E.
4/6	F F	Upper Case Letter F.
4/7	G G	Upper Case Letter G.
4/8	H H	Upper Case Letter H.
4/9	I I	Upper Case Letter I.
4/10	J J	Upper Case Letter J.
4/11	K K	Upper Case Letter K.
4/12	L L	Upper Case Letter L.
4/13	M M	Upper Case Letter M.
4/14	N N	Upper Case Letter N.

DATA TRANSMISSION CODES

1963, 1967 USASCII Characters (Cont)

COLUMN / ROW	CHARACTER	
	USASCII X3.4 - 1963 / USASCII X3.4 - 1967	NAME/FUNCTION
4/15	0 / 0	Upper Case Letter O.
5/0	P / P	Upper Case Letter P.
5/1	Q / Q	Upper Case Letter Q.
5/2	R / R	Upper Case Letter R.
5/3	S / S	Upper Case Letter S.
5/4	T / T	Upper Case Letter T.
5/5	U / U	Upper Case Letter U.
5/6	V / V	Upper Case Letter V.
5/7	W / W	Upper Case Letter W.
5/8	X / X	Upper Case Letter X.
5/9	Y / Y	Upper Case Letter Y.

1963, 1967 USASCII Characters (Cont)

COLUMN / ROW	CHARACTER	
	USASCII X3.4 - 1963 USASCII X3.4 - 1967	NAME/FUNCTION
5/10	z Z	Upper Case Letter Z.
5/11	[[	Opening Bracket.
5/12	\ \	Reverse Slant,
5/13	]]	Closing Bracket.
5/14	↑ ^	Exponentiation or Up Arrow. Circumflex.
5/15	← _	Replace by or Left Arrow. Underline.
6/0	 `	Unassigned. Grave Accent (Opening Single Quotation Mark).
6/1	 a	Unassigned. Lower Case Letter a.
6/2	 b	Unassigned. Lower Case Letter b.
6/3	 c	Unassigned. Lower Case Letter c.
6/4	 d	Unassigned. Lower Case Letter d.

DATA TRANSMISSION CODES

1963, 1967 USASCII Characters (Cont)

COLUMN / ROW	CHARACTER USASCII X3.4 - 1963 USASCII X3.4 - 1967	NAME/FUNCTION
6/5		Unassigned.
	e	Lower Case Letter e.
6/6		Unassigned.
	f	Lower Case Letter f.
6/7		Unassigned.
	g	Lower Case Letter g.
6/8		Unassigned.
	h	Lower Case Letter h.
6/9		Unassigned.
	i	Lower Case Letter i.
6/10		Unassigned.
	j	Lower Case Letter j.
6/11		Unassigned.
	k	Lower Case Letter k.
6/12		Unassigned.
	l	Lower Case Letter l.
6/13		Unassigned.
	m	Lower Case Letter m.
6/14		Unassigned.
	n	Lower Case Letter n.
6/15		Unassigned.
	o	Lower Case Letter o.
7/0		Unassigned.
	p	Lower Case Letter p.

1963, 1967 USASCII Characters (Cont)

COLUMN / ROW	CHARACTER: USASCII X3.4 - 1963 / USASCII X3.4 - 1967	CHARACTER: NAME/FUNCTION
7/1		Unassigned.
	q	Lower Case Letter q.
7/2		Unassigned.
	r	Lower Case Letter r.
7/3		Unassigned.
	s	Lower Case Letter s.
7/4		Unassigned.
	t	Lower Case Letter t.
7/5		Unassigned.
	u	Lower Case Letter u.
7/6		Unassigned.
	v	Lower Case Letter v.
7/7		Unassigned.
	w	Lower Case Letter w.
7/8		Unassigned.
	x	Lower Case Letter x.
7/9		Unassigned.
	y	Lower Case Letter y.
7/10		Unassigned.
	z	Lower Case Letter z.
7/11		Unassigned.
	{	Opening Brace.

DATA TRANSMISSION CODES

1963, 1967 USASCII Characters (Cont)

COLUMN / ROW	CHARACTER: USASCII X3.4 - 1963 / USASCII X3.4 - 1967	CHARACTER: NAME/FUNCTION
7/12	ACK	Acknowledge: A communication control character transmitted by a receiver as an affirmative response to a sender.
	| / |	Vertical Line.
7/13		Unassigned Control.
	}	Closing Brace.
7/14	ESC	Mode shift character used to indicate a departure from the standard set of basic characters; e.g., used to shift from upper to lower case letters.
	~	Overline (Tilde; General Accent)
7/15	DEL / DEL	Delete: This character is used primarily to "erase" or "obliterate" erroneous or unwanted characters in perforated tape. (In the strict sense, DEL is not a control character.)

Data Communications Control Codes

USASCII X3.4-1967 TRANSMISSION CONTROL CODES

Graphic	Function	Code 7654321
SOH	Start of heading	0000001
STX	Start of text	0000010
ETX	End of text	0000011
EOT	End of transmission	0000100
ENQ	Enquiry	0000101
ACK	Acknowledge	0000110
DLE	Data link escape	0010000
DC1	Device control	0010001
DC2	Device control	0010010
NAK	Negative acknowledge	0010101
SYN	Synchronous idle	0010100
ETB	End of transmission block	0010111
NUL	Null	0000000

ASCII X3.4-1963 TRANSMISSION CONTROL CODES

Graphic	Function	Code 7654321
SOM	Start of message	0000001
EOA	End of address	0000010
EOM	End of message	0000011
EOT	End of transmission	0000100
WRU	Who are you	0000101
RU	Are you	0000110
SYNC	Synchronous idle	0010110
ERR	Error	0010101
ACK	Acknowledge	1111100
DC1	(XON)	0010001
\	Reverse slash	1011100
*	Asterisk	0101010

PTTC/6 TRANSMISSION CONTROL CODES

Function	Code BA8421
EOT (end of transmission)	001111
EOA (end of address - #)	001011
EOB (end of block)	011110
Positive response-polling (#)	001011
Positive response-addressing (.)	111011
Positive response-inquiry (#)	001011
Negative response-polling (-)	100000
Negative response-addressing (-)	100000
Positive ACK-error control (.)	111011
Negative ACK-error control (-)	100000
Upper case	001110
Lower case	111110

PTTC/6 CONTROL CODE FUNCTIONS

Function	Code
Start of text	EOA
End of text	EOB
End of transmission	EOT
Response	.(period) or - (hyphen)
Negative response	- (hyphen)
Upper case shift	U.C.
Lower case shift	L.C.

DATA TRANSMISSION CODES

BAUDOT TRANSMISSION CONTROL CODES

Graphic	Code 54321
M	11100
V	11110
H	10100
S	00101
Figures	11011
Letters	11111

BAUDOT A CONTROL CODE FUNCTION

Function	Code
Start of text	Any code but B
End of text (read)	H
End of text (write)	H with a flag bit
Response	V or letters
Negative response	V
Upper case shift	Figures
Lower case shift	Letters
Enquiry	Letters

EBCDIC CONTROL CODES

ACK ACKNOWLEDGE - A communication control character transmitted by a receiver as an affirmative response to a sender.

BEL BELL - A character for use when there is a need to call for human attention. It may control alarm or attention devices.

BS BACKSPACE - A format effector which controls the movement of the printing position one printing space backward on the same printing line. (Applicable also to display devices.)

CAN CANCEL - A control character used to indicate that the data with which it is sent is in error or is to be disregarded.

CR CARRIAGE RETURN - A format effector which controls the movement of the printing position to the first printing position on the same printing line. (Applicable also to display devices.)

DC1 DC2 DC3 DC4 DEVICE CONTROLS - Characters for the control of ancillary devices associated with Data Processing or Telecommunication Systems, more especially switching devices "ON" or "OFF." (If a single "STOP" control is required to interrupt or turn off ancillary devices, DC4 is the preferred assignment.)

DEL DELETE - This character is used primarily to "ERASE" or "OBLITERATE" erroneous or unwanted characters in perforated tape. (In the strict sense, DEL is not a control character.)

EBCDIC CONTROL CODES (Cont)

DLE DATA LINK ESCAPE - A communication control character which will change the meaning of a limited number of contiguously following characters. It is used exclusively to provide supplementary controls in data communication networks.

EM END OF MEDIUM - A control character associated with the sent data which may be used to identify the physical end of the medium, or the end of the used, or wanted, portion of information recorded on a medium. (The position of this character does not necessarily correspond to the physical end of the medium.)

ENQ ENQUIRY - A communication control character used in Data Communication Systems as a request for a response from a Remote Station. It may be used as a "WHO YOU ARE" (WRU) to obtain identification, or may be used to obtain Station Status, or both.

EOT END OF TRANSMISSION - A communication control character used to indicate the conclusion of a transmission, which may have contained one or more texts and any associated headings.

ESC ESCAPE - A control character intended to provide code extension (supplementary characters) in General Information Interchange. The escape character itself is prefix affecting the interpretation of a limited number of contiguously following characters.

ETB END OF TRANSMISSION BLOCK - A communication control character used to indicate the end of a block of data for communication purposes. ETB is used for blocking data where the block structure is not necessarily related to the processing format.

ETX END OF TEXT - A communication control character used to terminate a sequence of characters started with STX and transmitted as an entity.

FF FORM FEED - A format effector which controls the movement of the printing position to the first predetermined printing line on the next form or page. (Applicable also to display devices.)

FS FILE SEPARATOR
GS GROUP SEPARATOR
RS RECORD SEPARATOR
US UNIT SEPARATOR

These information separators may be used within data in optional fashion, except that their hierarchical relationship shall be: FS is the most inclusive, then GS, then RS, and US is least inclusive. (The content and length of a File, Group, Record, or Unit are not specified.)

HT HORIZONTAL TABULATION - A format effector which controls the movement of the printing position to the next in a series of predetermined positions along the printing line. (Applicable also to display devices and the SKIP function on punched cards.)

DATA TRANSMISSION CODES

EBCDIC CONTROL CODES (Cont)

LF — LINE FEED - A format effector which controls the movement of the printing position to the next printing line. (Applicable also to display devices.)

NAK — NEGATIVE ACKNOWLEDGE - A communication control character transmitted by a receiver as a negative response to the sender.

NL — NEW LINE - A format effector which causes both Carriage Return and Line Feed.

NUL — NULL - The all-zeros character which may serve to accomplish time fill and media fill.

SI — SHIFT IN - A control character indicating that the code combinations which follow shall be interpreted according to the Standard Code Table.

SO — SHIFT OUT - A control character indicating that the code combinations which follow shall be interpreted as outside of the character set of the Standard Code Table until a shift in character is reached.

SOH — START OF HEADING - A communication control character used at the beginning of a sequence of characters which constitutes a machine-sensible address or routing information. Such a sequence is referred to as the "heading." An STX character has the effect of terminating a heading.

SP — SPACE - A normally non-printing graphic character used to separate words. It is also a format effector which controls the movement of the printing position, one printing position forward. (Applicable also to display devices.)

STX — START OF TEXT - A communication control character which precedes a sequence of characters that is to be treated as an entity and entirely transmitted through to the ultimate destination. Such a sequence is referred to as "TEXT." STX may be used to terminate a sequence of characters started by SOH.

SUB — SUBSTITUTE - A character that may be substituted for a character which is determined to be invalid or in error.

SYN — SYNCHRONOUS IDLE - A communication control character used by a synchronous transmission system in the absence of any other character to provide a signal from which synchronism may be achieved or retained.

TRANSLATION CHARTS

Two types of translation charts are presented; Multiplexor translation charts and Data Communications Processor translation charts. All other code translations must be accomplished programmatically.

MULTIPLEXOR TRANSLATION CHART
BCL TO EBCDIC

BCL Graphic	BCL External Hex. Code	EBCDIC Hex. Code	EBCDIC Graphic
?	00	6F	?
1	01	F1	1
2	02	F2	2
3	03	F3	3
4	04	F4	4
5	05	F5	5
6	06	F6	6
7	07	F7	7
8	08	F8	8
9	09	F9	9
0	0A	F0	0
#	0B	7B	#
:	0C	7A	:
@	0D	7C	@
>	0E	6E	>
≥	0F	7D	' (Apos)
Blank	10	40	SP
/	11	61	/
S	12	E2	S
T	13	E3	T
U	14	E4	U
V	15	E5	V
W	16	E6	W
X	17	E7	X
Y	18	E8	Y
Z	19	E9	Z
≠	1A	6D	_ (us)
,	1B	6B	,
%	1C	6C	%
=	1D	7E	=
]	1E	5A	]
"	1F	7F	"
-	20	60	-
J	21	D1	J
K	22	D2	K
L	23	D3	L
M	24	D4	M
N	25	D5	N
O	26	D6	O
P	27	D7	P
Q	28	D8	Q
R	29	D9	R
x (Mult)	2A	DO	MZ (I)
$	2B	5B	$
*	2C	5C	*
)	2D	5D	)
;	2E	5E	;
≤	2F	5F	¬
&	30	50	&
A	31	C1	A
B	32	C2	B
C	33	C3	C
D	34	C4	D
E	35	C5	E
F	36	C6	F
G	37	C7	G
H	38	C8	H
I	39	C9	I
+	3A	4E	+
.	3B	4B	.
[	3C	4A	[
(	3D	4D	(
<	3E	4C	<
÷	3F	4F	|

MULTIPLEXOR TRANSLATION CHART
EBCDIC TO BCL

EBCDIC Graphic	EBCDIC Hex. Code	BCL External Hex. Code	BCL Graphic
SP	40	10	Blank
[	4A	3C	[
.	4B	3B	.
<	4C	3E	<
(	4D	3D	(
+	4E	3A	+
|	4F	3F	+
&	50	30	&
]	5A	1E	]
$	5B	2B	$
*	5C	2C	*
)	5D	2D	)
;	5E	2E	;
¬	5F	2F	≤
-	60	20	-
/	61	11	/
,	6B	1B	,
%	6C	1C	%
_(us)	6D	1A	≠
>	6E	0E	>
?	6F	00	?
:	7A	0C	
#	7B	0B	#
@	7C	0D	@
'(apos)	7D	0F	≥
=	7E	1D	=
"	7F	1F	"
PZ(+)	C0	3A	+
A	C1	31	A
B	C2	32	B
C	C3	33	C
D	C4	34	D
E	C5	35	E
F	C6	36	F
G	C7	37	G
H	C8	38	H
I	C9	39	I
MZ(-)	D0	2A	x(Mult)
J	D1	21	J
K	D2	22	K
L	D3	23	L
M	D4	24	M
N	D5	25	N
O	D6	26	O
P	D7	27	P
Q	D8	28	Q
R	D9	29	R
S	E2	12	S
T	E3	13	T
U	E4	14	U
V	E5	15	V
W	E6	16	W
X	E7	17	X
Y	E8	18	Y
Z	E9	19	Z
0	F0	0A	0
1	F1	01	1
2	F2	02	2
3	F3	03	3
4	F4	04	4
5	F5	05	5
6	F6	06	6
7	F7	07	7
8	F8	08	8
9	F9	09	9

DCP TRANSLATION CHART
EBCDIC TO USASCII

EBCDIC	USASCII	EBCDIC	USASCII	EBCDIC	USASCII	EBCDIC	USASCII	EBCDIC	USASCII	EBCDIC	USASCII	EBCDIC	USASCII	EBCDIC	USASCII
00	00 NUL	10	10 DLE	20	1	30	1	40	20 SP	50	26 &	60	2D -	70	1
01	01 SOH	11	11 DC1	21	1	31	1	41	1	51	1	61	2F /	71	1
02	02 STX	12	12 DC2	22	1	32	16 SYN	42	1	52	1	62	1	72	1
03	03 ETX	13	13 DC3	23	1	33	1	43	1	53	1	63	1	73	1
04	1	14	1	24	1	34	1	44	1	54	1	64	1	74	1
05	09 HT	15	1	25	0A LF	35	1	45	1	55	1	65	1	75	1
06	1	16	08 BS	26	17 ETB	36	1	46	1	56	1	66	1	76	1
07	7F DEL	17	1	27	1B ESC	37	04 EOT	47	1	57	1	67	1	77	1
08	1	18	18 CAN	28	1	38	1	48	1	58	1	68	1	78	1
09	1	19	19 EM	29	1	39	1	49	1	59	1	69	1	79	60 \
0A	1	1A	1	2A	1	3A	1	4A	5B [	5A	5D]	6A	7C ¦	7A	3A :
0B	0B VT	1B	1	2B	1	3B	1	4B	2E .	5B	24 $	6B	2C ,	7B	23 #
0C	0C FF	1C	1C FS	2C	1	3C	14 DC4	4C	3C <	5C	2A *	6C	25 %	7C	40 @
0D	0D CR	1D	1D GS	2D	05 ENQ	3D	15 NAK	4D	28 (	5D	29)	6D	5F _	7D	27 '
0E	0E SO	1E	1E RS	2E	06 ACK	3E	1	4E	2B +	5E	3B ;	6E	3E >	7E	3D =
0F	0F SI	1F	1F US	2F	07 BEL	3F	1A SUB	4F	21 !	5F	5E ¬	6F	3F ?	7F	22 "

1 EBCDIC input codes without correspondence in USASCII are translated to USASCII 00 NULL

DCP TRANSLATION CHART
EBCBIC TO USASCII (Cont)

EBCDIC	USASCII	EBCDIC	USASCII	EBCDIC	USASCII	EBCDIC	USASCII	EBCDIC	USASCII	EBCDIC	USASCII	EBCDIC	USASCII	EBCDIC	USASCII
80	1	90	1	A0	1	B0	1	C0	7B {	D0	7D }	E0	5C \	F0	30 0
81	61 a	91	6A j	A1	7E ~	B1	1	C1	41 A	D1	4A J	E1	1	F1	31 1
82	62 b	92	6B k	A2	73 s	B2	1	C2	42 B	D2	4B K	E2	53 S	F2	32 2
83	63 c	93	6C l	A3	74 t	B3	1	C3	43 C	D3	4C L	E3	54 T	F3	33 3
84	64 d	94	6D m	A4	75 u	B4	1	C4	44 D	D4	4D M	E4	55 U	F4	34 4
85	65 e	95	6E n	A5	76 v	B5	1	C5	45 E	D5	4E N	E5	56 V	F5	35 5
86	66 f	96	6F o	A6	77 w	B6	1	C6	46 F	D6	4F O	E6	57 W	F6	36 6
87	67 g	97	70 p	A7	78 x	B7	1	C7	47 G	D7	50 P	E7	58 X	F7	37 7
88	68 h	98	71 q	A8	79 y	B8	1	C8	48 H	D8	51 Q	E8	59 Y	F8	38 8
89	69 i	99	72 r	A9	7A z	B9	1	C9	49 I	D9	52 R	E9	5A Z	F9	39 9
8A	1	9A	1	AA	1	BA	1	CA	1	DA	1	EA	1	FA	1
8B	1	9B	1	AB	1	BB	1	CB	1	DB	1	EB	1	FB	1
8C	1	9C	1	AC	1	BC	1	CC	1	DC	1	EC	1	FC	1
8D	1	9D	1	AD	1	BD	1	CD	1	DD	1	ED	1	FD	1
8E	1	9E	1	AE	1	BE	1	CE	1	DE	1	EE	1	FE	1
8F	1	9F	1	AF	1	BF	1	CF	1	DF	1	EF	1	FF	1

1 EBCDIC input codes without correspondence in USASCII are translated to USASCII 00 NULL

DCP TRANSLATION CHART

EBCDIC TO BCL

EBCDIC	BCL	EBCDIC	BCL	EBCDIC	BCL	EBCDIC	BCL	EBCDIC	BCL	EBCDIC	BCL	EBCDIC	BCL	EBCDIC	BCL
00	1	10	1	20	1	30	1	40	10 SP	50	30 &	60	20 -	70	1
01	1	11	1	21	1	31	1	41	1	51	1	61	11 /	71	1
02	1	12	1	22	1	32	1	42	1	52	1	62	1	72	1
03	1	13	1	23	1	33	1	43	1	53	1	63	1	73	1
04	1	14	1	24	1	34	1	44	1	54	1	64	1	74	1
05	1	15	1	25	1	35	1	45	1	55	1	65	1	75	1
06	1	16	1	26	1	36	1	46	1	56	1	66	1	76	1
07	1	17	1	27	1	37	1	47	1	57	1	67	1	77	1
08	1	18	1	28	1	38	1	48	1	58	1	68	1	78	1
09	1	19	1	29	1	39	1	49	1	59	1	69	1	79	1
0A	1	1A	1	2A	1	3A	1	4A	3C [	5A	1E]	6A	1	7A	0D :
0B	1	1B	1	2B	1	3B	1	4B	3B .	5B	2B $	6B	1B ,	7B	0B #
0C	1	1C	1	2C	1	3C	1	4C	3E <	5C	2C *	6C	1C %	7C	0C @
0D	1	1D	1	2D	1	3D	1	4D	3D (	5D	2D)	6D	1	7D	1
0E	1	1E	1	2E	1	3E	1	4E	3A +	5E	2E ;	6E	0E >	7E	1D ■
0F	1	1F	1	2F	1	3F	1	4F	1	5F	1	6F	00 ?	7F	1F □

1 EBCDIC input codes without correspondence in BCL are translated to BCL 00 ? (question mark)

PRINTED IN U.S. AMERICA
5000276

DCP TRANSLATION CHART
EBCDIC TO BCL (Cont)

EBCDIC	BCL	EBCDIC	BCL	EBCDIC	BCL	EBCDIC	BCL	EBCDIC	BCL	EBCDIC	BCL	EBCDIC	BCL	EBCDIC	BCL
80	1	90	1	A0	1	B0	1	C0	3A +	D0	2A ×	E0	1	F0	0A 0
81	1	91	1	A1	1	B1	1	C1	31 A	D1	21 J	E1	1	F1	01 1
82	1	92	1	A2	1	B2	1	C2	32 B	D2	22 K	E2	12 S	F2	02 2
83	1	93	1	A3	1	B3	1	C3	33 C	D3	23 L	E3	13 T	F3	03 3
84	1	94	1	A4	1	B4	1	C4	34 D	D4	24 M	E4	14 U	F4	04 4
85	1	95	1	A5	1	B5	1	C5	35 E	D5	25 N	E5	15 V	F5	05 5
86	1	96	1	A6	1	B6	1	C6	36 F	D6	26 O	E6	16 W	F6	06 6
87	1	97	1	A7	1	B7	1	C7	37 G	D7	27 P	E7	17 X	F7	07 7
88	1	98	1	A8	1	B8	1	C8	38 H	D8	28 Q	E8	18 Y	F8	08 8
89	1	99	1	A9	1	B9	1	C9	39 I	D9	29 R	E9	19 Z	F9	09 9
8A	1	9A	1	AA	1	BA	1	CA	1	DA	1	EA	1	FA	1
8B	1	9B	1	AB	1	BB	1	CB	1	DB	1	EB	1	FB	1
8C	1	9C	1	AC	1	BC	1	CC	1	DC	1	EC	1	FC	1
8D	1	9D	1	AD	1	BD	1	CD	1	DD	1	ED	1	FD	1
8E	1	9E	1	AE	1	BE	1	CE	1	DE	1	EE	1	FE	1
8F	1	9F	1	AF	1	BF	1	CF	1	DF	1	EF	1	FF	1

1 EBCDIC input codes without correspondence in BCL are translated to BCL 00 ? (question mark)

DCP TRANSLATION CHART
USASCII TO EBCDIC

USASCII	EBCDIC	USASCII	EBCDIC	USASCII	EBCDIC	USASCII	EBCDIC	USASCII	EBCDIC	USASCII	EBCDIC	USASCII	EBCDIC	USASCII	EBCDIC
00	00 NUL	10	10 DLE	20	40 SP	30	F0 0	40	7C @	50	D7 P	60	79 \	70	97 p
01	01 SOH	11	11 DC1	21	4F !	31	F1 1	41	C1 A	51	D8 Q	61	81 a	71	98 q
02	02 STX	12	12 DC2	22	7F "	32	F2 2	42	C2 B	52	D9 R	62	82 b	72	99 r
03	03 ETX	13	13 DC3	23	7B #	33	F3 3	43	C3 C	53	E2 S	63	83 c	73	A2 s
04	37 ECT	14	3C DC4	24	5B $	34	F4 4	44	C4 D	54	E3 T	64	84 d	74	A3 t
05	2D ENQ	15	3D NAK	25	6C %	35	F5 5	45	C5 E	55	E4 U	65	85 e	75	A4 u
06	2E ACK	16	32 SYN	26	50 &	36	F6 6	46	C6 F	56	E5 V	66	86 f	76	A5 v
07	2F BEL	17	26 ETB	27	7D '	37	F7 7	47	C7 G	57	E6 W	67	87 g	77	A6 w
08	16 BS	18	18 CAN	28	4D (	38	F8 8	48	C8 H	58	E7 X	68	88 h	78	A7 x
09	05 HT	19	19 EM	29	5D)	39	F9 9	49	C9 I	59	E8 Y	69	89 i	79	A8 y
0A	25 LF	1A	3F SUB	2A	5C *	3A	7A :	4A	D1 J	5A	E9 Z	6A	91 j	7A	A9 z
0B	0B VT	1B	27 ESC	2B	4E +	3B	5E ;	4B	D2 K	5B	4A [	6B	92 k	7B	C0 {
0C	0C FF	1C	1C FS	2C	6B ,	3C	4C <	4C	D3 L	5C	E0 \	6C	93 l	7C	6A ¦
0D	0D CR	1D	1D GS	2D	60 -	3D	7E =	4D	D4 M	5D	5A]	6D	94 m	7D	D0 }
0E	0E SO	1E	1E RS	2E	4B .	3E	6E >	4E	D5 N	5E	5F ^	6E	95 n	7E	A1 ~
0F	0F SI	1F	1F US	2F	61 /	3F	6F ?	4F	D6 O	5F	6D _	6F	96 o	7F	07 DEL

NOTE: All other input codes (80 through FF) are translated to EBCDIC 00 NULL

TRANSLATION CHARTS

DCP TRANSLATION CHART
BCL TO EBCDIC

BCL	EBCDIC		BCL	EBCDIC		BCL	EBCDIC		BCL	EBCDIC	
00	6F	?	10	40	SP	20	60	-	30	50	&
01	F1	1	11	61	/	21	D1	J	31	C1	A
02	F2	2	12	E2	S	22	D2	K	32	C2	B
03	F3	3	13	E3	T	23	D3	L	33	C3	C
04	F4	4	14	E4	U	24	D4	M	34	C4	D
05	F5	5	15	E5	V	25	D5	N	35	C5	E
06	F6	6	16	E6	W	26	D6	O	36	C6	F
07	F7	7	17	E7	X	27	D7	P	37	C7	G
08	F8	8	18	E8	Y	28	D8	Q	38	C8	H
09	F9	9	19	E9	Z	29	D9	R	39	C9	I
0A	F0	0	1A	6F(1)	?	2A	D0(2)	)	3A	4E	+
0B	7B	#	1B	6B	,	2B	5B	$	3B	4B	.
0C	7C	@	1C	6C	%	2C	5C	*	3C	4A	[
0D	7A	:	1D	7E	=	2D	5D	)	3D	4D	(
0E	6E	>	1E	5A	]	2E	5E	;	3E	4C	<
0F	6F(1)	?	1F	7F	"	2F	6F(1)	?	3F	6F(1)	?

NOTE: All other input codes (40 through FF) are translated to EBCDIC 00 NUL

1 BCL input codes without correspondence in EBCDIC are translated to EBCDIC 6F ? (question mark)

2 Denotes "minus zero"

SECTION I

PROCESSOR OPERATORS

The terms "Multiplexor" and "I/O Processor" are synonymous.

SECTION 1

PROCESSOR OPERATORS

The hexadecimal value of the code is preceded by a P, V, U, or E. All codes preceded by a P (primary) are one to seven syllables long. All codes preceded by a V (variant) are two syllables long, the first syllable being hexadecimal 95 (Escape to 16 Bit Instruction). All codes preceded by a U (unconditional) can be used as one syllable codes, or as two syllable codes when preceded by hexadecimal 95. All codes preceded by an E (edit) are Edit micro-operators.

If the hexadecimal code is followed by a C, the operator is only valid in control state. In the case of Scan Out, only some functions are restricted to control state.

PROCESSOR OPERATORS, BY MNEMONICS

MNEMONIC	HEXADECIMAL CODE	NAME
ADD	P80	ADD
BRFL	PA0	BRANCH FALSE
BRST	P9E	BIT RESET
BRTR	PA1	BRANCH TRUE
BRUN	PA2	BRANCH UNCONDITIONAL
BSET	P96	BIT SET
CBON	V8B	COUNT BINARY ONES
CEQD	PF4	COMPARE CHARACTERS EQUAL, DESTRUCTIVE
CEQU	PFC	COMPARE CHARACTERS EQUAL, UPDATE
CGED	PF1	COMPARE CHARACTERS GREATER OR EQUAL, DESTRUCTIVE
CGEU	PF9	COMPARE CHARACTERS GREATER OR EQUAL, UPDATE
CGTD	PF2	COMPARE CHARACTERS GREATER, DESTRUCTIVE
CGTU	PFA	COMPARE CHARACTERS GREATER, UPDATE
CHSN	P8E	CHANGE SIGN BIT
CLED	PF3	COMPARE CHARACTERS LESS OR EQUAL, DESTRUCTIVE
CLEU	PFB	COMPARE CHARACTERS LESS OR EQUAL, UPDATE
CLSD	PF0	COMPARE CHARACTERS LESS, DESTRUCTIVE
CLSU	PF8	COMPARE CHARACTERS LESS, UPDATE
CNED	PF5	COMPARE CHARACTERS NOT EQUAL, DESTRUCTIVE
CNEU	PFD	COMPARE CHARACTERS NOT EQUAL, UPDATE
DBFL	PA8	DYNAMIC BRANCH FALSE
DBRS	P9F	DYNAMIC BIT RESET
DBST	P97	DYNAMIC BIT SET
DBTR	PA9	DYNAMIC BRANCH TRUE
DBUN	PAA	DYNAMIC BRANCH UNCONDITIONAL
DEXI	V47	DISABLE EXTERNAL INTERRUPTS
DFTR	P99	DYNAMIC FIELD TRANSFER
DINS	P90	DYNAMIC FIELD INSERT
DISO	P9B	DYNAMIC FIELD ISOLATE
DIVD	P83	DIVIDE

PROCESSOR OPERATORS

PROCESSOR OPERATORS, BY MNEMONICS (Cont)

MNEMONIC	HEXADECIMAL CODE	NAME
DLET	P85	DELETE TOP OF STACK
DSLF	PC1	DYNAMIC SCALE LEFT
DSRF	PC7	DYNAMIC SCALE RIGHT FINAL
DSRR	PC9	DYNAMIC SCALE RIGHT ROUND
DSRS	PC5	DYNAMIC SCALE RIGHT SAVE
DSRT	PC3	DYNAMIC SCALE RIGHT TRUNCATE
DUPL	P87	DUPLICATE TOP OF STACK
EEXI	V46	ENABLE EXTERNAL INTERRUPTS
ENDE	EDE	END EDIT
ENDF	ED5	END FLOAT
ENTR	PAB	ENTER
EQUL	P8C	EQUAL
EVAL	PAC	EVALUATE DESCRIPTOR
EXCH	PB6	EXCHANGE
EXIT	PA3	EXIT
EXPU	P0D	EXECUTE SINGLE MICRO, SINGLE POINTER UPDATE
EXSD	PD2	EXECUTE SINGLE MICRO, DESTRUCTIVE
EXSU	PDA	EXECUTE SINGLE MICRO, UPDATE
FLTR	P9B	FIELD TRANSFER
GREQ	P89	GREATER THAN OR EQUAL
GRTR	P8A	GREATER THAN
HALT	UDF	CONDITIONAL HALT
HEYU	V4F	INTERRUPT OTHER PROCESSORS
ICVD	PCA	INPUT CONVERT, DESTRUCTIVE
ICVU	PC8	INPUT CONVERT, UPDATE
IDIV	P84	INTEGER DIVIDE
IDLE	V44	IDLE UNTIL INTERRUPT
IMKS	PCF	INSERT MARK STACK
INDX	PA6	INDEX
INOP	ED8	INSERT OVERPUNCH
INSC	EDD	INSERT CONDITIONAL
INSG	ED9	INSERT DISPLAY SIGN
INSR	P9C	FIELD INSERT
INSU	EDC	INSERT UNCONDITIONAL
(IRWL)	VAD	(HARDWARE PSEUDO-INSTRUCTION)
ISOL	P9A	FIELD ISOLATE
JOIN	V42	SET TWO SINGLES TO DOUBLE
LAND	P90	LOGICAL AND
LEQV	P93	LOGICAL EQUIVALENCE
LESS	P88	LESS THAN
LLLU	V8D	LINKED LIST LOOKUP
LNOT	P92	LOGICAL NEGATE
LOAD	P8D	LOAD
LODT	V8C	LOAD TRANSPARENT
LOG2	V8B	LEADING ONE TEST

PROCESSOR OPERATORS, BY MNEMONICS (Cont)

MNEMONIC	HEXADECIMAL CODE	NAME
LOR	P91	LOGICAL OR
LSEQ	P8B	LESS THAN OR EQUAL
LT16	PB3	LIT CALL 16 BITS
LT48	PBE	LIT CALL 48 BITS
LT8	PB2	LIT CALL 8 BITS
MCHR	E07	MOVE CHARACTERS
MFLT	E01	MOVE WITH FLOAT
MINS	EDO	MOVE WITH INSERT
MKST	PAE	MARK STACK
MPCW	PBF	MAKE PROGRAM CONTROL WORD
MULT	P82	MULTIPLY
MULX	P8F	EXTENDED MULTIPLY
MVNU	ED6	MOVE NUMERIC UNCONDITIONAL
MVST	VAF	MOVE TO STACK
NAMC	P40=>7F	NAME CALL
NEQL	P8D	NOT EQUAL
NOOP	UFE	NO OPERATION
NTGD	V87	INTEGERIZE, ROUNDED, DOUBLE PRECISION
NTGR	P87	INTEGERIZE, ROUNDED
NTIA	P86	INTEGERIZE, TRUNCATED
NVLD	UFF	INVALID OPERATOR
NXLN	PA5	INDEX AND LOAD NAME
NXLV	PAO	INDEX AND LOAD VALUE
OCRX	V85	OCCURS INDEX
ONE	PB1	LIT CALL ONE
OVRD	PBA	OVERWRITE DESTRUCTIVE
OVRN	PB8	OVERWRITE NON-DESTRUCTIVE
PACD	PD1	PACK DESTRUCTIVE
PACU	PD9	PACK UPDATE
(PCWL)	VAE	(HARDWARE PSEUDO-INSTRUCTION)
PUSH	P84	PUSH DOWN STACK REGISTERS
RDIV	P85	REMAINDER DIVIDE
RDLK	VBA	READ WITH LOCK
RETN	PA7	RETURN
ROFF	PD7	READ AND CLEAR OVERFLOW FLIP-FLOP
RPRR	V88	READ PROCESSOR REGISTER
RSDN	V87	ROTATE STACK DOWN
RSIF	E04	RESET FLOAT
RSUP	VB6	ROTATE STACK UP
RTAG	V85	READ TAG FIELD
RTFF	PDE	READ TRUE/FALSE FLIP-FLOP
SAME	P94	LOGICAL EQUAL
SCLF	PCO	SCALE LEFT
SCNI	V4A	SCAN IN
SCNO	V4B C	SCAN OUT
SCRF	PC6	SCALE RIGHT FINAL

PROCESSOR OPERATORS

PROCESSOR OPERATORS, BY MNEMONICS (Cont)

MNEMONIC	HEXADECIMAL CODE	NAME
SCRR	PC8	SCALE RIGHT ROUND
SCRS	PC4	SCALE RIGHT SAVE
SCRT	PC2	SCALE RIGHT TRUNCATE
SEQD	VF4	SCAN WHILE EQUAL, DESTRUCTIVE
SEQU	VFC	SCAN WHILE EQUAL, UPDATE
SFDC	EDA	SKIP FORWARD DESTINATION CHARACTERS
SFSC	E02	SKIP FORWARD SOURCE CHARACTERS
SGED	VF1	SCAN WHILE GREATER OR EQUAL, DESTRUCTIVE
SGEU	VF9	SCAN WHILE GREATER OR EQUAL, UPDATE
SGTD	VF2	SCAN WHILE GREATER, DESTRUCTIVE
SGTU	VFA	SCAN WHILE GREATER, UPDATE
SINT	V45 C	SET INTERVAL TIMER
SISO	P05	STRING ISOLATE
SLED	VF3	SCAN WHILE LESS OR EQUAL, DESTRUCTIVE
SLEU	VFB	SCAN WHILE LESS OR EQUAL, UPDATE
SLSD	VF0	SCAN WHILE LESS, DESTRUCTIVE
SLSU	VF8	SCAN WHILE LESS, UPDATE
SNED	VF5	SCAN WHILE NOT EQUAL, DESTRUCTIVE
SNEU	VFD	SCAN WHILE NOT EQUAL, UPDATE
SNGL	PCD	SET TO SINGLE PRECISION, ROUNDED
SNGT	PCC	SET TO SINGLE PRECISION, TRUNCATED
SPLT	V43	SET DOUBLE TO TWO SINGLES
SPRR	VB9	SET PROCESSOR REGISTER
SRCH	VBE	MASKED SEARCH FOR EQUAL
SRDC	EDB	SKIP REVERSE DESTINATION CHARACTERS
SRSC	E03	SKIP REVERSE SOURCE CHARACTERS
STAG	VB4	SET TAG FIELD
STBR	PA4	STEP AND BRANCH
STFF	PAF	STUFF ENVIRONMENT
STOD	PB8	STORE DESTRUCTIVE
STON	PB9	STORE NON-DESTRUCTIVE
SUBT	PB1	SUBTRACT
SWFD	VD4	SCAN WHILE FALSE, DESTRUCTIVE
SWFU	VDC	SCAN WHILE FALSE, UPDATE
SWTD	VD5	SCAN WHILE TRUE, DESTRUCTIVE
SWTU	VDD	SCAN WHILE TRUE, UPDATE
SXSN	P06	SET EXTERNAL SIGN
TEED	PD0	TABLE ENTER EDIT, DESTRUCTIVE
TEEU	PD8	TABLE ENTER EDIT, UPDATE
TEQD	PE4	TRANSFER WHILE EQUAL, DESTRUCTIVE
TEQU	PEC	TRANSFER WHILE EQUAL, UPDATE
TGED	PE1	TRANSFER WHILE GREATER OR EQUAL, DESTRUCTIVE
TGEU	PE9	TRANSFER WHILE GREATER OR EQUAL, UPDATE
TGTD	PE2	TRANSFER WHILE GREATER, DESTRUCTIVE
TGTU	PEA	TRANSFER WHILE GREATER, UPDATE

PROCESSOR OPERATORS, BY MNEMONICS (Cont)

MNEMONIC	HEXADECIMAL CODE	NAME
TLED	PE3	TRANSFER WHILE LESS OR EQUAL, DESTRUCTIVE
TLEU	PEB	TRANSFER WHILE LESS OR EQUAL, UPDATE
TLSD	PE0	TRANSFER WHILE LESS, DESTRUCTIVE
TLSU	PE8	TRANSFER WHILE LESS, UPDATE
TNED	PE5	TRANSFER WHILE NOT EQUAL, DESTRUCTIVE
TNEU	PED	TRANSFER WHILE NOT EQUAL, UPDATE
TRNS	VD7	TRANSLATE
TUND	PE6	TRANSFER UNCONDITIONAL, DESTRUCTIVE
TUNU	PEE	TRANSFER UNCONDITIONAL, UPDATE
TWFD	VD2	TRANSFER WHILE FALSE, DESTRUCTIVE
TWFU	VDA	TRANSFER WHILE FALSE, UPDATE
TWOD	PD4	TRANSFER WORDS OVERWRITE DESTRUCTIVE
TWOU	PDC	TRANSFER WORDS OVERWRITE UPDATE
TWSD	PD3	TRANSFER WORDS, DESTRUCTIVE
TWSU	PDB	TRANSFER WORDS UPDATE
TWTD	VD3	TRANSFER WHILE TRUE, DESTRUCTIVE
TWTU	VDB	TRANSFER WHILE TRUE, UPDATE
UABD	VD1	UNPACK ABSOLUTE, DESTRUCTIVE
UABU	VD9	UNPACK ABSOLUTE, UPDATE
USND	VD0	UNPACK SIGNED, DESTRUCTIVE
USNU	VD8	UNPACK SIGNED, UPDATE
VALC	P00=>3F	VALUE CALL
VARI	P95	ESCAPE TO 16-BIT INSTRUCTION
WHOI	V4E	READ PROCESSOR IDENTIFICATION
XTND	PCE	SET TO DOUBLE PRECISION
ZERO	PB0	LIT CALL ZERO

PROCESSOR OPERATORS, BY NAME

NAME	MNEMONIC	HEXADECIMAL CODE
ADD	ADD	PB0
BIT RESET	BRST	P9E
BIT SET	BSET	P96
BRANCH FALSE	BRFL	PA0
BRANCH TRUE	BRTR	PA1
BRANCH UNCONDITIONAL	BRUN	PA2
CHANGE SIGN BIT	CHSN	P8E
COMPARE CHARACTERS EQUAL, DESTRUCTIVE	CEQD	PF4
COMPARE CHARACTERS EQUAL, UPDATE	CEQU	PFC
COMPARE CHARACTERS GREATER OR EQUAL, DESTRUCTIVE	CGED	PF1

PROCESSOR OPERATORS

PROCESSOR OPERATORS, BY NAME (Cont)

NAME	MNEMONIC	HEXADECIMAL CODE
COMPARE CHARACTERS GREATER OR EQUAL, UPDATE	CGEU	PF9
COMPARE CHARACTERS GREATER, DESTRUCTIVE	CGTD	PF2
COMPARE CHARACTERS GREATER, UPDATE	CGTU	PFA
COMPARE CHARACTERS LESS OR EQUAL, DESTRUCTIVE	CLED	PF3
COMPARE CHARACTERS LESS OR EQUAL, UPDATE	CLEU	PFB
COMPARE CHARACTERS LESS, DESTRUCTIVE	CLSD	PF0
COMPARE CHARACTERS LESS, UPDATE	CLSU	PF8
COMPARE CHARACTERS NOT EQUAL, DESTRUCTIVE	CNED	PF5
COMPARE CHARACTERS NOT EQUAL, UPDATE	CNEU	PFD
CONDITIONAL HALT	HALT	UDF
COUNT BINARY ONES	CBON	VBB
DELETE TOP OF STACK	DLET	PB5
DISABLE EXTERNAL INTERRUPTS	DEXI	V47
DIVIDE	DIVD	P83
DUPLICATE TOP OF STACK	DUPL	PB7
DYNAMIC BIT RESET	DBRS	P9F
DYNAMIC BIT SET	DBST	P97
DYNAMIC BRANCH FALSE	DBFL	PA8
DYNAMIC BRANCH TRUE	DBTR	PA9
DYNAMIC BRANCH UNCONDITIONAL	DBUN	PAA
DYNAMIC FIELD INSERT	DINS	P9D
DYNAMIC FIELD ISOLATE	DISO	P9B
DYNAMIC FIELD TRANSFER	DFTR	P99
DYNAMIC SCALE LEFT	DSLF	PC1
DYNAMIC SCALE RIGHT FINAL	DSRF	PC7
DYNAMIC SCALE RIGHT ROUND	DSRR	PC9
DYNAMIC SCALE RIGHT SAVE	DSRS	PC5
DYNAMIC SCALE RIGHT TRUNCATE	DSRT	PC3
ENABLE EXTERNAL INTERRUPTS	EEXI	V46
END EDIT	ENDE	EDE
END FLOAT	ENDF	ED5
ENTER	ENTR	PA8
EQUAL	EQUL	P8C
ESCAPE TO 16-BIT INSTRUCTION	VARI	P95
EVALUATE DESCRIPTOR	EVAL	PAC
EXCHANGE	EXCH	PB6
EXECUTE SINGLE MICRO, SINGLE POINTER UPDATE	EXPU	P0D
EXECUTE SINGLE MICRO, DESTRUCTIVE	EXSD	P02
EXECUTE SINGLE MICRO, UPDATE	EXSU	P0A
EXIT	EXIT	PA3
EXTENDED MULTIPLY	MULX	P8F
FIELD INSERT	INSR	P9C
FIELD ISOLATE	ISOL	P9A
FIELD TRANSFER	FLTR	P9B
GREATER THAN	GRTR	PBA

PROCESSOR OPERATORS, BY NAME (Cont)

NAME	MNEMONIC	HEXADECIMAL CODE
GREATER THAN OR EQUAL	GREQ	P89
(HARDWARE PSEUDO-INSTRUCTION)	(RWL)	VAD
(HARDWARE PSEUDO-INSTRUCTION)	(PCWL)	VAE
IDLE UNTIL INTERRUPT	IDLE	V44
INDEX	INDX	PA6
INDEX AND LOAD NAME	NXLN	PA5
INDEX AND LOAD VALUE	NXLV	PAD
INPUT CONVERT, DESTRUCTIVE	ICVD	PCA
INPUT CONVERT, UPDATE	ICVU	PCB
INSERT CONDITIONAL	INSC	EDD
INSERT DISPLAY SIGN	INSG	ED9
INSERT MARK STACK	(MKS	PCF
INSERT OVERPUNCH	INOP	ED8
INSERT UNCONDITIONAL	INSU	EDC
INTEGER DIVIDE	IDIV	P84
INTEGERIZE, ROUNDED	NTGR	P87
INTEGERIZE, ROUNDED, DOUBLE PRECISION	NTGD	V87
INTEGERIZE, TRUNCATED	NTIA	P86
INTERRUPT OTHER PROCESSORS	HEYU	V4F
INVALID OPERATOR	NVLD	UFF
LEADING ONE TEST	LOG2	V88
LESS THAN	LESS	P88
LESS THAN OR EQUAL	LSEQ	P8B
LINKED LIST LOOKUP	LLLU	VBD
LIT CALL ONE	ONE	PB1
LIT CALL ZERO	ZERO	PB0
LIT CALL 16 BITS	LIT6	PB3
LIT CALL 48 BITS	LIT48	PBE
LIT CALL 8 BITS	LIT8	PB2
LOAD	LOAD	PBD
LOAD TRANSPARENT	LODT	VBC
LOGICAL AND	LAND	P90
LOGICAL EQUAL	SAME	P94
LOGICAL EQUIVALENCE	LEQV	P93
LOGICAL NEGATE	LNOT	P92
LOGICAL OR	LOR	P91
MAKE PROGRAM CONTROL WORD	MPCW	PBF
MARK STACK	MKST	PAE
MASKED SEARCH FOR EQUAL	SRCH	VBE
MOVE CHARACTERS	MCHR	ED7
MOVE NUMERIC UNCONDITIONAL	MVNU	ED6
MOVE TO STACK	MVST	VAF
MOVE WITH FLOAT	MFLT	ED1
MOVE WITH INSERT	MINS	EDD
MULTIPLY	MULT	P82

PROCESSOR OPERATORS

PROCESSOR OPERATORS, BY NAME (Cont)

NAME	MNEMONIC	HEXADECIMAL CODE
NAME CALL	NAMC	P40=>7F
NO OPERATION	NOOP	UFE
NOT EQUAL	NEQL	PBD
OCCURS INDEX	OCRX	VB5
OVERWRITE DESTRUCTIVE	OVRD	PBA
OVERWRITE NON-DESTRUCTIVE	OVRN	PBB
PACK DESTRUCTIVE	PACD	PD1
PACK UPDATE	PACU	PD9
PUSH DOWN STACK REGISTERS	PUSH	PB4
READ AND CLEAR OVERFLOW FLIP-FLOP	ROFF	PD7
READ PROCESSOR IDENTIFICATION	WHOI	V4E
READ PROCESSOR REGISTER	RPRR	VB8
READ TAG FIELD	RTAG	VB5
READ TRUE/FALSE FLIP-FLOP	RTFF	PDE
READ WITH LOCK	RDLK	VBA
REMAINDER DIVIDE	RDIV	PB5
RESET FLOAT	RSTF	ED4
RETURN	RETN	PA7
ROTATE STACK DOWN	RSDN	VB7
ROTATE STACK UP	RSUP	VB6
SCALE LEFT	SCLF	PC0
SCALE RIGHT FINAL	SCRF	PC6
SCALE RIGHT ROUND	SCRR	PC8
SCALE RIGHT SAVE	SCRS	PC4
SCALE RIGHT TRUNCATE	SCRT	PC2
SCAN IN	SCNI	V4A
SCAN OUT	SCNO	V4B C
SCAN WHILE EQUAL, DESTRUCTIVE	SEQD	VF4
SCAN WHILE EQUAL, UPDATE	SEQU	VFC
SCAN WHILE FALSE, DESTRUCTIVE	SWFD	VD4
SCAN WHILE FALSE, UPDATE	SWFU	VDC
SCAN WHILE GREATER OR EQUAL, DESTRUCTIVE	SGED	VF1
SCAN WHILE GREATER OR EQUAL, UPDATE	SGEU	VF9
SCAN WHILE GREATER, DESTRUCTIVE	SGTD	VF2
SCAN WHILE GREATER, UPDATE	SGTU	VFA
SCAN WHILE LESS OR EQUAL, DESTRUCTIVE	SLED	VF3
SCAN WHILE LESS OR EQUAL, UPDATE	SLEU	VFB
SCAN WHILE LESS, DESTRUCTIVE	SLSD	VF0
SCAN WHILE LESS, UPDATE	SLSU	VF8
SCAN WHILE NOT EQUAL, DESTRUCTIVE	SNED	VF5
SCAN WHILE NOT EQUAL, UPDATE	SNEU	VFD
SCAN WHILE TRUE, DESTRUCTIVE	SWTD	VD5
SCAN WHILE TRUE, UPDATE	SWTU	VDD
SET DOUBLE TO TWO SINGLES	SPLT	V43
SET EXTERNAL SIGN	SXSN	PD6

PRINTED IN U.S. AMERICA 5000276

PROCESSOR OPERATORS, BY NAME (Cont)

NAME	MNEMONIC	HEXADECIMAL CODE
SET INTERVAL TIMER	SINT	V45 C
SET PROCESSOR REGISTER	SPRR	VB9
SET TAG FIELD	STAG	VB4
SET TO DOUBLE PRECISION	XTND	PCE
SET TO SINGLE PRECISION, ROUNDED	SNGL	PCD
SET TO SINGLE PRECISION, TRUNCATED	SNGT	PCC
SET TWO SINGLES TO DOUBLE	JOIN	V42
SKIP FORWARD DESTINATION CHARACTERS	SFDC	EDA
SKIP FORWARD SOURCE CHARACTERS	SFSC	ED2
SKIP REVERSE DESTINATION CHARACTERS	SRDC	EDB
SKIP REVERSE SOURCE CHARACTERS	SRSC	ED3
STEP AND BRANCH	STBR	PA4
STORE DESTRUCTIVE	STOD	PB8
STORE NON-DESTRUCTIVE	STON	PB9
STRING ISOLATE	SISO	PD5
STUFF ENVIRONMENT	STFF	PAF
SUBTRACT	SUBT	PB1
TABLE ENTER EDIT, DESTRUCTIVE	TEED	PD0
TABLE ENTER EDIT, UPDATE	TEEU	PD8
TRANSFER UNCONDITIONAL, DESTRUCTIVE	TUND	PE6
TRANSFER UNCONDITIONAL, UPDATE	TUNU	PEE
TRANSFER WHILE EQUAL, DESTRUCTIVE	TEQD	PE4
TRANSFER WHILE EQUAL, UPDATE	TEQU	PEC
TRANSFER WHILE FALSE, DESTRUCTIVE	TWFD	V02
TRANSFER WHILE FALSE, UPDATE	TWFU	VDA
TRANSFER WHILE GREATER OR EQUAL, DESTRUCTIVE	TGED	PE1
TRANSFER WHILE GREATER OR EQUAL, UPDATE	TGEU	PE9
TRANSFER WHILE GREATER, DESTRUCTIVE	TGTD	PE2
TRANSFER WHILE GREATER, UPDATE	TGTU	PEA
TRANSFER WHILE LESS OR EQUAL, DESTRUCTIVE	TLED	PE3
TRANSFER WHILE LESS OR EQUAL, UPDATE	TLEU	PEB
TRANSFER WHILE LESS, DESTRUCTIVE	TLSD	PED
TRANSFER WHILE LESS, UPDATE	TLSU	PE8
TRANSFER WHILE NOT EQUAL, DESTRUCTIVE	TNED	PE5
TRANSFER WHILE NOT EQUAL, UPDATE	TNEU	PED
TRANSFER WHILE TRUE, DESTRUCTIVE	TWTD	VD3
TRANSFER WHILE TRUE, UPDATE	TWTU	VDB
TRANSFER WORDS OVERWRITE DESTRUCTIVE	TWOD	PD4
TRANSFER WORDS OVERWRITE UPDATE	TWOU	PDC
TRANSFER WORDS, DESTRUCTIVE	TWSD	PD3
TRANSFER WORDS, UPDATE	TWSU	PD8
TRANSLATE	TRNS	V07
UNPACK ABSOLUTE, DESTRUCTIVE	UABD	VD1
UNPACK ABSOLUTE, UPDATE	UABU	VD9
UNPACK SIGNED, DESTRUCTIVE	USND	VDO
UNPACK SIGNED, UPDATE	USNU	VD8
VALUE CALL	VALC	PDO=>3F

PROCESSOR OPERATORS, BY HEXADECIMAL CODE

HEXADECIMAL CODE	NAME	MNEMONIC
P00=>3F	VALUE CALL	VALC
P40=>7F	NAME CALL	NAMC
V42	SET TWO SINGLES TO DOUBLE	JOIN
V43	SET DOUBLE TO TWO SINGLES	SPLT
V44	IDLE UNTIL INTERRUPT	IDLE
V45	SET INTERVAL TIMER	SINT
V46	ENABLE EXTERNAL INTERRUPTS	EEXI
V47	DISABLE EXTERNAL INTERRUPTS	DEXI
V4A	SCAN IN	SCNI
V4B	SCAN OUT	SCNO
V4E	READ PROCESSOR IDENTIFICATION	WHOI
V4F	INTERRUPT OTHER PROCESSORS	HEYU
P80	ADD	ADD
P81	SUBTRACT	SUBT
P82	MULTIPLY	MULT
P83	DIVIDE	DIVD
P84	INTEGER DIVIDE	IDIV
P85	REMAINDER DIVIDE	RDIV
V85	OCCURS INDEX	OCRX
P86	INTEGERIZE, TRUNCATED	NTIA
P87	INTEGERIZE, ROUNDED	NTGR
V87	INTEGERIZE, ROUNDED, DOUBLE PRECISION	NTGD
P88	LESS THAN	LESS
P89	GREATER THAN OR EQUAL	GREQ
P8A	GREATER THAN	GRTR
P8B	LESS THAN OR EQUAL	LSEQ
V8B	LEADING ONE TEST	LOG2
P8C	EQUAL	EQUL
P8D	NOT EQUAL	NEQL
P8E	CHANGE SIGN BIT	CHSN
P8F	EXTENDED MULTIPLY	MULX
P90	LOGICAL AND	LAND
P91	LOGICAL OR	LOR
P92	LOGICAL NEGATE	LNOT
P93	LOGICAL EQUIVALENCE	LEQV
P94	LOGICAL EQUAL	SAME
P95	ESCAPE TO 16-BIT INSTRUCTION	VARI
P96	BIT SET	BSET
P97	DYNAMIC BIT SET	DBST
P98	FIELD TRANSFER	FLTR
P99	DYNAMIC FIELD TRANSFER	DFTR
P9A	FIELD ISOLATE	ISOL
P9B	DYNAMIC FIELD ISOLATE	DISO
P9C	FIELD INSERT	INSR
P9D	DYNAMIC FIELD INSERT	DINS

PRINTED IN U.S. AMERICA 5000276

PROCESSOR OPERATORS, BY HEXADECIMAL CODE (Cont)

HEXADECIMAL CODE	NAME	MNEMONIC
P9E	BIT RESET	BRST
P9F	DYNAMIC BIT RESET	DBRS
PA0	BRANCH FALSE	BRFL
PA1	BRANCH TRUE	BRTR
PA2	BRANCH UNCONDITIONAL	BRUN
PA3	EXIT	EXIT
PA4	STEP AND BRANCH	STBR
PA5	INDEX AND LOAD NAME	NXLN
PA6	INDEX	INDX
PA7	RETURN	RETN
PA8	DYNAMIC BRANCH FALSE	DBFL
PA9	DYNAMIC BRANCH TRUE	DBTR
PAA	DYNAMIC BRANCH UNCONDITIONAL	DBUN
PAB	ENTER	ENTR
PAC	EVALUATE DESCRIPTOR	EVAL
PAD	INDEX AND LOAD VALUE	NXLV
VAD	(HARDWARE PSEUDO-OPERATOR)	(IRWL)
PAE	MARK STACK	MKST
VAE	(HARDWARE PSEUDO-OPERATOR)	(PCWL)
PAF	STUFF ENVIRONMENT	STFE
VAF	MOVE TO STACK	MVST
PB0	LIT CALL ZERO	ZERO
PB1	LIT CALL ONE	ONE
PB2	LIT CALL 8 BITS	LT8
PB3	LIT CALL 16 BITS	LT16
PB4	PUSH DOWN STACK REGISTERS	PUSH
VB4	SET TAG FIELD	STAG
PB5	DELETE TOP OF STACK	DLET
VB5	READ TAG FIELD	RTAG
PB6	EXCHANGE	EXCH
VB6	ROTATE STACK UP	RSUP
PB7	DUPLICATE TOP OF STACK	DUPL
VB7	ROTATE STACK DOWN	RSDN
PB8	STORE DESTRUCTIVE	STOD
VB8	READ PROCESSOR REGISTER	RPRR
PB9	STORE NON-DESTRUCTIVE	STON
VB9	SET PROCESSOR REGISTER	SPRR
PBA	OVERWRITE DESTRUCTIVE	OVRD
VBA	READ WITH LOCK	RDLK
PBB	OVERWRITE NON-DESTRUCTIVE	OVRN
VBB	COUNT BINARY ONES	CBON
VBC	LOAD TRANSPARENT	LODT
PBD	LOAD	LOAD
VBD	LINKED LIST LOOKUP	LLLU
PBE	LIT CALL 48 BITS	LT48

PROCESSOR OPERATORS

PROCESSOR OPERATORS, BY HEXADECIMAL CODE (Cont)

HEXADECIMAL CODE	NAME	MNEMONIC
VBE	MASKED SEARCH FOR EQUAL	SRCH
PBF	MAKE PROGRAM CONTROL WORD	MPCW
PC0	SCALE LEFT	SCLF
PC1	DYNAMIC SCALE LEFT	DSLF
PC2	SCALE RIGHT TRUNCATE	SCRT
PC3	DYNAMIC SCALE RIGHT TRUNCATE	DSRT
PC4	SCALE RIGHT SAVE	SCRS
PC5	DYNAMIC SCALE RIGHT SAVE	DSRS
PC6	SCALE RIGHT FINAL	SCRF
PC7	DYNAMIC SCALE RIGHT FINAL	DSRF
PC8	SCALE RIGHT ROUND	SCRR
PC9	DYNAMIC SCALE RIGHT ROUND	DSRR
PCA	INPUT CONVERT, DESTRUCTIVE	ICVD
PCB	INPUT CONVERT, UPDATE	ICVU
PCC	SET TO SINGLE PRECISION, TRUNCATED	SNGT
PCD	SET TO SINGLE PRECISION, ROUNDED	SNGL
PCE	SET TO DOUBLE PRECISION	XTND
PCF	INSERT MARK STACK	IMKS
EB0	MOVE WITH INSERT	MINS
PD0	TABLE ENTER EDIT, DESTRUCTIVE	TEED
VD0	UNPACK SIGNED, DESTRUCTIVE	USND
ED1	MOVE WITH FLOAT	MFLT
PD1	PACK DESTRUCTIVE	PACD
VD1	UNPACK ABSOLUTE, DESTRUCTIVE	UABD
ED2	SKIP FORWARD SOURCE CHARACTERS	SFSC
PD2	EXECUTE SINGLE MICRO, DESTRUCTIVE	EXSD
VD2	TRANSFER WHILE FALSE, DESTRUCTIVE	TWFD
ED3	SKIP REVERSE SOURCE CHARACTERS	SRSC
PD3	TRANSFER WORDS, DESTRUCTIVE	TWSD
VD3	TRANSFER WHILE TRUE, DESTRUCTIVE	TWTD
ED4	RESET FLOAT	RSTF
PD4	TRANSFER WORD, OVERWRITE DESTRUCTIVE	TWOD
VD4	SCAN WHILE FALSE, DESTRUCTIVE	SWFD
ED5	END FLOAT	ENDF
PD5	STRING ISOLATE	SISO
VD5	SCAN WHILE TRUE, DESTRUCTIVE	SWTD
ED6	MOVE NUMERIC UNCONDITIONAL	MVNU
PD6	SET EXTERNAL SIGN	SXSN
ED7	MOVE CHARACTERS	MCHR
PD7	READ AND CLEAR OVERFLOW FLIP-FLOP	ROFF
VD7	TRANSLATE	TRNS
ED8	INSERT OVERPUNCH	INOP
PD8	TABLE ENTER EDIT, UPDATE	TEEU
VD8	UNPACK SIGNED UPDATE	USNU
ED9	INSERT DISPLAY SIGN	INSG

PRINTED IN U.S. AMERICA 5000276

PROCESSOR OPERATORS, BY HEXADECIMAL CODE (Cont)

HEXADECIMAL CODE	NAME	MNEMONIC
PD9	PACK UPDATE	PACU
VD9	UNPACK ABSOLUTE, UPDATE	UABU
EDA	SKIP FORWARD DESTINATION CHARACTERS	SFDC
PDA	EXECUTE SINGLE MICRO, UPDATE	EXSU
VDA	TRANSFER WHILE FALSE, UPDATE	TWFU
EDB	SKIP REVERSE DESTINATION CHARACTERS	SRDC
PDB	TRANSFER WORDS, UPDATE	TWSU
VDB	TRANSFER WHILE TRUE, UPDATE	TWTU
EDC	INSERT UNCONDITIONAL	INSU
PDC	TRANSFER WORDS OVERWRITE UPDATE	TWOU
VDC	SCAN WHILE FALSE, UPDATE	SWFU
EDD	INSERT CONDITIONAL	INSC
PDD	EXECUTE SINGLE MICRO, SINGLE POINTER UPDATE	EXPU
VDD	SCAN WHILE TRUE, UPDATE	SWTU
EDE	END EDIT	ENDE
PDE	READ TRUE/FALSE FLIP-FLOP	RTFF
UDF	CONDITIONAL HALT	HALT
PE0	TRANSFER WHILE LESS, DESTRUCTIVE	TLSD
PE1	TRANSFER WHILE GREATER OR EQUAL, DESTRUCTIVE	TGED
PE2	TRANSFER WHILE GREATER, DESTRUCTIVE	TGTD
PE3	TRANSFER WHILE LESS OR EQUAL, DESTRUCTIVE	TLED
PE4	TRANSFER WHILE EQUAL, DESTRUCTIVE	TEQD
PE5	TRANSFER WHILE NOT EQUAL, DESTRUCTIVE	TNED
PE6	TRANSFER UNCONDITIONAL, DESTRUCTIVE	TUND
PE8	TRANSFER WHILE LESS, UPDATE	TLSU
PE9	TRANSFER WHILE GREATER OR EQUAL, UPDATE	TGEU
PEA	TRANSFER WHILE GREATER, UPDATE	TGTU
PEB	TRANSFER WHILE LESS OR EQUAL, UPDATE	TLEU
PEC	TRANSFER WHILE EQUAL, UPDATE	TEQU
PE0	TRANSFER WHILE NOT EQUAL, UPDATE	TNEU
PEE	TRANSFER UNCONDITIONAL, UPDATE	TUNU
PF0	COMPARE CHARACTERS LESS, DESTRUCTIVE	CLSD
VF0	SCAN WHILE LESS, DESTRUCTIVE	SLSD
PF1	COMPARE CHARACTERS GREATER OR EQUAL, DESTRUCTIVE	CGED
VF1	SCAN WHILE GREATER OR EQUAL, DESTRUCTIVE	SGED
PF2	COMPARE CHARACTERS GREATER, DESTRUCTIVE	CGTD
VF2	SCAN WHILE GREATER, DESTRUCTIVE	SGTD
PF3	COMPARE CHARACTERS LESS OR EQUAL, DESTRUCTIVE	CLED
VF3	SCAN WHILE LESS OR EQUAL, DESTRUCTIVE	SLED
PF4	COMPARE CHARACTERS EQUAL, DESTRUCTIVE	CEQD
VF4	SCAN WHILE EQUAL, DESTRUCTIVE	SEQD
PF5	COMPARE CHARACTERS NOT EQUAL, DESTRUCTIVE	CNED
VF5	SCAN WHILE NOT EQUAL, DESTRUCTIVE	SNED
PF8	COMPARE CHARACTERS LESS, UPDATE	CLSU
VF8	SCAN WHILE LESS, UPDATE	SLSU

PROCESSOR OPERATORS

PROCESSOR OPERATORS, BY HEXADECIMAL CODE (Cont)

HEXADECIMAL CODE	NAME	MNEMONIC
PF9	COMPARE CHARACTERS GREATER OR EQUAL, UPDATE	CGEU
VF9	SCAN WHILE GREATER OR EQUAL, UPDATE	SGEU
PFA	COMPARE CHARACTERS GREATER, UPDATE	CGTU
VFA	SCAN WHILE GREATER, UPDATE	SGTU
PFB	COMPARE CHARACTERS LESS OR EQUAL, UPDATE	CLEU
VFB	SCAN WHILE LESS OR EQUAL, UPDATE	SLEU
PFC	COMPARE CHARACTERS EQUAL, UPDATE	CEQU
VFC	SCAN WHILE EQUAL, UPDATE	SEQU
PFD	COMPARE CHARACTERS NOT EQUAL, UPDATE	CNEU
VFD	SCAN WHILE NOT EQUAL, UPDATE	SNEU
UFE	NO OPERATION	NOOP
UFF	INVALID OPERATOR	NVLD

FUNCTIONAL DESCRIPTION OF OPERATORS

This table provides a shorthand description of the action of each operator. The table provides for each operator stack conditions necessary to initialize the operation, a quick description of the action of each operator, the resultant stack conditions, and a reference to the appropriate hardware flow chart. Stack conditions are indicated by: a dash (-), neither register used; a zero (0), register empty; a one (1), register full or valid; and a two (2), this register ignored.

Legend of symbols used:

←	replaced by or loaded from	^	logical "and"
@	addressed by	/	divide
::	as defined by	;	then
[ ]	contents of	↔	exchanged with
M	memory	(OC)	operation complete
>	greater than	(e)	evaluate
<	less than	MBLB	memory protect bit
>>	right shifted by	ICM	IC memory
<<	left shifted by	M	main core memory
≥	greater or equal	(PsC)→	pseudo call on
≤	less than or equal	MSD	most significant digit
=	equal	—▷	left justified
≠	not equal	—▷	right justified
×	times	( )	contents within are repeated
+	arithmetic plus	(8=)	bit equivalent
-	arithmetic minus	AAd	address adder
Δ	complement of	Cr	"C" register
and	English conjunction	CR	Specific bit of "C" register
or	logical "or"	rC	"C" register

FAMILIES F, G, H GLOSSARY OF UNIQUE NAMES

ALOW	allow memory protect to interrupt
CHPL	conditional halt level
COMT	compare type operators
DERR	descriptor error
DFRJ	displace from right justified character position
DIB	destination character pointer register
DIC	destination character pointer, taken in reverse order
DPRF	destination pointer read only flip-flop
DREL	destination pointer required
DSZ	destination size register
DTRJ	displace to right justified character position
EELX	E-register equals load x
EEZL	E-register equals zero or MADF
EXPF	execute single pointer flip-flop
EXPL	execute single pointer level
EXSF	execute single shot flip-flop
EXTF	external sign flip-flop
FINI	finish of end edit
INBK	inhibit back up
MOIF	move with insert or move with float
MNMC	move numeric unconditional or move characters
NGLL	≠, <,> operators
NINO	not "insert overpunch"
NVLO	invalid operator level, and part of interrupt handler
NVLF	invalid flip-flop (for pseudo call NVLO)
P2R2	binary two stored in P2 interrupt parameter
P2R3	binary three stored in P2 interrupt parameter
P2R4	binary four stored in P2 interrupt parameter
QFO1	invalid operator interrupt flip-flop
QFO2	presence bit interrupt flip-flop
QFO3	memory protected interrupt flip-flop
QFO4	segmented array interrupt flip-flop
RNTR	returning to table edit mode from interrupt
SEQD	source character equals dest. char.
SERR	error resulting from compare test
SIB	source character pointer register

PROCESSOR OPERATORS

FAMILIES F, G, H GLOSSARY OF UNIQUE NAMES (CONT)

SIC	source character pointer, taken in reverse order	TEEL	table enter edit level
SOPF	source pointer equals an operand	TFOF	true/false occupied flip-flop
SOTW	scan or transfer while compare	UABL	unpack absolute level
SSZ	source size register	USNL	unpack signed level
TEEF	table enter edit flip-flop	UPDF	update flip-flop
		XROF	X register is occupied

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
P00←3F	VALC	C	Value Call	0	2	Dnn ← [syllable one], MAnn = [Dnn] + [syllable two], Ar←M@MAnn. If A [50:3] = 0X0, then AROF←1 If A[50:3] = 100, then A[50:35]←0 and AROF←1 If A[50:3] = 001, then Ar←M@IRW If A[50:3] = 101, then Ar←M@D.D.	1	2	C4
P3F←7F	NAMC	C	Name Call	0	2	Dnn = [syllable one], MAnn = [Dnn] + [syllable two], Ar←MAnn, A[50:3]←001, and AROF←1.	1	2	C56
V42	JOIN	C	Set two singles to double	1	1	Ar←Yr, AROF←0, and BR49←1	0	1	C53
V43	SPLT	C	Set double to two singles	0	1	Yr←Ar, BR49←0, Yr49←0, and AROF←1	1	1	C53
V44	IDLE	C	Idle till interrupt	-	-	IIHF←0. Processor operation suspended until (EXTI, PTPI, or ITIS)←1	-	-	C55
V45	SINT	C	Set interval timer	1	2	Ar is integerized. Interval timer←A [10:11]	0	2	C52

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				ARØF	BRØF		ARØF	BRØF	
V46	EEXI	C	Enable external interrupts	-	-	NCSF←0 and IIHF←0	-	-	C55
V47	DEXI	C	Disable external interrupts	-	-	IIHF←1 and NCSF←1	-	-	C55
V4A	SCNI	C	Scan In	1	0	Br←MPX::[Ar]	0	1	C57
V4B	SCNO	C	Scan out	1	1	Br←MPX::[Ar]	0	0	C59
V4E	WHOI	C	Read processor identification	0	2	Processor identification←A[2:3]	1	2	C54
V4F	MEYU	C	Interrupt other processor	-	-	PTP1←1	-	-	C54
P80	ADD	A	Add	1	1	Br←Ar+Br or BrYr←ArXr+BrYr if AR49 = 1 and/or BR49 = 1	0	1	A3
P81	SUBT	A	Subtract	1	1	Br←Br-Ar or BrYr←BrYr-ArXr if AR49 = 1 and/or BR49 = 1	0	1	A3
P82	MULT	A	Multiply	1	1	Br←BrxAr or BrYr←BrYrxArXr if AR49 = 1 and/or BR49 = 1	0	1	A15

PRINTED IN U.S. AMERICA
5000274

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
P83	DIVD	A	Divide	1	1	Br←Br/Ar or BrYr+BrYr/ArXr if AR49 = 1 and/or BR49 = 1	0	1	A24
P84	IDIV	A	Integer divide	1	1	Br←Br/Ar until B[44:6] = 0 or BrYr←BrYr/ArXr until B[44:6] = 13 and Y[47:9] = 0 when AR49 = 1 and/or BR49 = 1.	0	1	A24
P85	RDIV	A	Remainder divide	1	1	Br/Ar until B[44:6] =0 and Br←BB. BrYr/ArXr until B[44:6] = 13 + Y[47:9] = 0 and Br←BB when AR49 = 1 and/or BR49 = 1	0	1	A24
V85	OCRX	A	Occurs index	1	1	B[15:16]←{(B[15:16]-1)xA[47:16]}+ A[15:16]	0	1	A51
P86	NTIA	A	Integerize, truncated	0	1	Br<<by B[44:6] when BR46 = 0 until B[44:6] = 0 Br>>by B[44:6] when BR46 = 1 until B[44:6] = 0	0	1	A42
P87	NTGR	A	Integerize, rounded	0	1	Br << by B[44:6] when BR46 = 0, until B[44:6] = 0 Br >> by B[44:6] when BR46 = 1, until B[46:6] = 0 Br←[Br] + 1 when Y[38:3]≥4	0	1	A42

PRINTED IN U.S. AMERICA
5000276

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
V87	NTGD	A	Integerize, rounded, double precision	0	1	BR49←1 if BR49=0. BrYr >> until B[44:6] = 13 + Y[47:9] = 0 when BR46 = 1 or BR46 = 0 + B[44:6]<13 BrYr << until B[44:6] = 13 and Y[47:9] = 0 when BR46 = 0 and B[44:6]>13 BrYr←BrYr + 1 if XR38 = 1	0	1	A47
P88	LESS	A	Less than	1	1	BR00←1 and B[50:50]←0 if Br<Ar B[50:51]←0 if Br≥Ar	0	1	A38
P89	GREQ	A	Greater than equal	1	1	BR00←1 and B[50:50]←0 if Br≥Ar B[50:51]←0 if Br<Ar	0	1	A38
P8A	GRTR	A	Greater than	1	1	BR00←1 and B[50:50]←0 if Br>Ar B[50:50]←0 if Br≤Ar	0	1	A38
P8B	LSEQ	A	Less than or equal	1	1	BR00←1 and B[50:50]←0 if Br≤Ar B[50:51]←0 if Br>Ar	0	1	A38
V8B	LOG2	A	Leading one test	0	1	Br←$(8mn + 1)^8$	0	1	A53
P8C	EQUL	A	Equal	1	1	BR00←1 and B[50:50]←0 if Br = Ar B[50:51]←0 if Br ≠ Ar	0	1	A38
P8D	NEQL	A	Not equal	1	1	BR00←1 and B[50:50]←0 if Br ≠ Ar B[50:51]←0 if Br = Ar	0	1	A38

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
P8E	CHSN	A	Change sign bit	1	2	AR46←ΔAR46	1	2	A46
P8F	MULX	A	Extended multiply	1	1	BrYr←ArxBr and BR49←1 if BR49 = 0 BrYr←ArXrxBrYr if BR49 = 1	0	1	A15
P90	LAND	B	Logical and	1	1	Brnn←1 when BRnn = 1 and ARnn = 1 if AR49 = 1 or BR49 = 1 then YRnn←1 when XRnn = 1 and YRnn = 1	0	1	B1
P91	LOR	B	Logical or	1	1	BRnn←1 when BRnn = 1 and/or Arnn = 1. If AR49 = 1 or BR49 = 1, then YRnn←1 when YRnn = 1 and/or XRnn = 1	0	1	B1
P92	LNOT	B	Logical negate	1	2	A[47:48]←ΔA[47:48] when BR49 = 0 A[47:48]←ΔA[47:48] and X[47:48]←ΔX[47:48] when BR49 = 1	1	2	B3
P93	LEQV	B	Logical equivalence	1	1	BRnn←1 when BRnn = ARnn	0	1	B1
P94	SAME	B	Logical equal	1	1	BR00←1 and B[50:50]←0 when A[50:51] = B[50:51] B[50:51]←0 when A[50:51] ≠ B[50:51]. If AR49 = 1 or BR49 = 1; BR00←1 and B[50:50]←0 when A[50:51] = B[50:51] and X[50:51] = Y[50:51] or B[50:51]←0 if A[50:51] ≠ B[50:52] or X[50:52] ≠ Y[50:52]	0	1	B2

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
P95	VARI	-	Escape to 16-bit instruction	-	-	VARF←1	-	-	P1
P96	BSET	B	Bit set	1	0	K = [second syllable]. TOA←K, TOM←(K-1), DIS←(D-K), and BROO←1. ARnn←BROO::TOA, TOM, and DIS	1	0	B17
P97	DBST	B	Dynamic bit set	0	1	Integerized value of Br = K. TOA←K, TOM←(K-1),	1	0	B19
				1	0	DIS←(D-K), and BRDO←1. ARnn←BROO::TOA, TOM and DIS			
P98	FLTR	B	Field transfer	1	1	K = [second syllable], G = [third syllable], L = [fourth syllable]. TOA←K, TOM←K+L, and DIS←(G-K). Br←Ar::TOA, TOM, and DIS	0	1	B4
P99	DFTR	B	Dynamic field tranafer	0	1	Integerized value of Br = L. Br←Cr	0	1	B6
				0	1	Integerized value of Br = G. ACR←AA and Br←Cr			
				0	1	Integerized value of Br = K. Br←BB			
				1	1	TOA←K, TOM←(K-L), and DIS←(G-K). Br←Ar::TOA, TOM, and DIS			

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
P9A	ISOL	B	Field isolate	1	0	G = [second syllable] and L = [third syllable]. TOA←(L-1), TOM←47, and DIS←(G+1-L). Br←Ar::TOA, TOM, and DIS	0	1	B13
P9B	DISO	B	Dynamic field	0	1	Integerized value of Br = L. C←(L-1)	1	0	B15
			Isolate	0	1	Integerized value of Br = G.			
				0	1	TOA←(L-1), TOM←47, and DIS←(G+1-L). Ar←Br::TOA, TOM, and DIS			
P9C	INSR	B	Field Insert	1	1	Second syllable = K, third syllable = L. TOA←K, TOM←(K-L), and DIS←(L-1-K). Br←Ar::TOA, TOM, and DIS	0	1	B9
P9D	DINS	B	Dynamic field	1	1	Integerized value of Br = L. Br→Cr	0	1	B11
			Insert	0	1	Integerized value of Br = K.			
				0	1	TOA←K, TOM←(K-L), and DIS←(L-1-K). Br←Ar::TOA, TOM, and DIS			
P9E	BRST	B	Bit reset	1	0	Second syllable = K:TOA←K, TOM←(K-1), DIS←(0-K) and BR00←0. ARnn←BR00::TOA, TOM, and DIS	1	0	B17

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
P9F	DBRS	B	Dynamic bit reset	0 1	1 0	Integerized value Br = K, TOA+K, TOM+(K-1), DIS+(0-K), and BROO+0. Arnn+BROO::TOA, TOM, and DIS	0	1	B19
PA0	BRFL	C	Branch false	1	2	[Second syllable]→C[7:8], PSR+C[7:3], C[7:8]→C[15:8], [Third syllable]→C[7:8], PIR+C[12:13], MA+[PIR]+[PBR], and Pr+M[MA if AROO = 0 PSR+PSR+3 and ADJ(0,1) if AROO = 1	0	2	C46
PA1	BRTR	C	Branch true	1	2	[Second syllable]→C[7:9], PSR+C[7:3], C[7:8]→C[15:8], [Third syllable]→C[7:8], PIR+C[12:13], MA+[PIR]+[PBR], and Pr+M[MA if AROO = 1, PSR+(PSR+3) and ADJ(0,1) if AROO = 0	0	2	C46
PA2	BRUN	C	Branch unconditional	-	-	[Second syllable]→C[7:8], PSR+C[7:3], C[7:8]→C[15:8], [Third syllable]→C[7:8], PIR+C[12:13], MA+[PIR]+[PBR], and Pr+M[MA	-	-	C46
PA3	EXIT	C	Exit	0	0	Cr+M[[F]+1, PSR+C[35:3], LL+C[18:5], PDR+C[13:14], S+[BOSR]+1, BOS Interrupt+0 = [F] - [BOSR]+1, Cr+M[[F], F+[F] - C[13:14], BUF+[F] - C[13:14]. SUFL+1 when [TEMP - [BUF]≥0. Cr+M[[BUF] and PBR+C[19:20]	0	0	C30

PRINTED IN U.S. AMERICA
5000276

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
PA4	STBR	C	Step and branch	1	2	If A[50:3] = 001; Cr←M@Ar, If A[50:3] = 101; Cr←M@Ar, If C[50:3] = 0 x 0; Cr←Ar, ADJ(0,1) and Ar←0 If C[50:3] = 001; Cr←Ar If C[50:3] = 101; Cr←Ar If C[50:3] = 100; TEMP←C[19:20], A[29:10]←C[47:10], TEMP←[TEMP]+A[39:20], C[19:20]←BUF2, M←Cr@[BUF2], QC6F←1 If CVF>FVF. PSR←[PSR]+3 If QC6F = 0 If QC6F = 1; syllable two [7:3]→PSR, PIR←syllable two [4:5]+ syllable three, and PROF←0	0	2	C1
PA5	NXLN	C	Index and load name	1	1	If A[50:3] = 0x0; Ar↔Br, Cr←Ar, and AROF←1 If A[50:3] = 001; Ar←Cr, Cr←M@Ar If A[50:3] = 101 and AR45 = 1; Invalid operator interrupt occurs. If A[50:3] = 101 and AR45 = 0; integerized value of B[19:20]→A[19:20], AR45←1, AR46←1, Cr←M@Ar. If C[50:3] = 101 and AR45 = 1; Invalid operator interrupt occurs. If C[50:3] = 101 and AR45 ≠ 0; Ar←Cr	1	0	C1

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
PA6	INDX	C	Index	1	1	If A[50:3] = 0x0; Ar↔Br, Cr←Ar and AROF←1 If A[50:3] = 001; Ar←Cr, Cr←M@Ar If A[50:3] = 101 and AR45 = 1; invalid operator interrupt occurs; IF A[50:3] = 101 and AR45 = 0; integerized value of B[19:20]←A[19:20], AR45←1 and AR46←1	1	0	C1
PA7	RETN	C	Return	0	1	Cr←M@[F]+1, PSR←C[35:3], LL←C[18:5], PBR←C[13:14],	0	1	C30
				1	2	S←F-1, BOS [interrupt←0 = [F] - [BOSR]+1, Cr←M@[F]; F←[F] - C[13:14], BUF←[F] - C[13:14]. SUFL←1 when [TEMP] - [BUF]≥0,Cr←M@[BUF] and PBR←C[19:20]. If B[50:30] = 0x0 or 100; operation is complete. If B[50:3] ≠ 0x0 or 100; ADJ[1,2], Cr←M@Ar until C[50:3] = 0x0 or 101)	1	1	

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
PA8	DBFL	C	Dynamic branch false	1	1	If BR00 = 1; QC If BR00 = 0 and A[50:3] = 0x0; Ar is integerized, PSR←0 if AR00 = 0 or PSR←3 if AR00 = 1, A[20:20]→PIR, and PROF←0. If BR00 = 0 and A[50:3] = 001; Cr←M@Ar If BR00 = 0, A[50:3] = 111; sequence error Interrupt←A[18:5] ≠ LL, [PSR, PIR, and PDR]←[PCW] if A[18:5] = LL	0	0	C45
PA9	DBTR	C	Dynamic branch true	1	1	If BR00 = 0; QC If BR00 = 1 and A[50:3] = 0x0; Ar is integerized, PSR←0 if AR00 = 0 or PSR←3 if AR00 = 1, A[20:20]→PIR and PROF←0 If BR00 = 1 and A[50:3] = 001; Cr←M@Ar If BR00 = 1 and A[50:3] = 111; sequence error Interrupt←A[18:5] ≠ LL, [PSR, PIR PDR]←[PCW] if [18:5] = LL	0	0	C41

PRINTED IN U.S. AMERICA
5000276

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
PAA	DBUN	C	Dynamic branch unconditional	1	2	If A[50:3] = 0x0; Ar is Integerized, PSR←0 if ARDO = 0 or PSR←3 if ARDO = 1, A[20:20]→PIR and PROF←0. If A[50:3] = 001; Cr←M@Ar If A[50:3] = 111; sequence error Interrupt←A[18:5] ≠ LL, or [PSR, PIR, PDR]←[PCW] if [18:5] = LL	0	0	C45
PAB	ENTR	C	Enter	0	0	Cr←M@[F]+1. Invalid operator Interrupt←C[50:3] ≠ 001. Ar←Cr, Cr←M@Ar until C[50:3] = 111, Xr←Cr, C[50:3]←011, [Cr]←(PDR, PSR, PIR, NCSF, and LL), (PDR, PSR, PIR, AND LL)←[Xr], M←[Cr]@[F]+1, Cr←M@[F], C[45:10]←A[45:10], C[35:10]←A[35:10], C[18:5]←LL, M←Cr@[F], DLL←F, Cr←M@[PDR], Sequence error Interrupt←C[50:3] ≠ 011, PBR←C[19:20] if C[50:3] = 111, Pr←M@[PIR] + [PBR], and (OC)	0	0	C3

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
PAC	EVAL	C	Evaluate descriptor	1	2	Cr←M@Ar and Ar←Cr until C[50:3] = 0x0 or A[50:3] = 101.	1	2	C2
PAD	NXLV	C	Index and load value	1	1	If A[50:3] = 0x0; Ar←→Br, Cr←Ar and ARDF←1 If A[50:3] = 001; Ar←Cr, Cr←M@Ar. If A[50:3] = 101 and AR45 = 1; Invalid interrupt occurs. If A[50:3] = 101 and AR45 = 0; integerized value of B[19:20]→A[19:20], AR45←1, AR46←1, and Cr←M@Ar If C[50:3] ≠ 0x0; Invalid operator interrupt occurs. If C[50:3] = 0x0;	1	0	C1
VAD	IRWL	C	Hardware pseudo operator	-	-	Ar←Cr, Cr←M@Ar	-	-	C5
PAE	MKST	C	Mark stack	0	0	TEMP←[S]+1, TEMP←TEMP - F, F←TEMP + F, B[13:14]←TEMP[13:14], B[50:3]←011, BRDF←1, ADJ(0,0), and (RC)	0	0	C50

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
VAE	PCWL	C	Hardware pseudo operator	-	-	BXPI = 1, Xr←Cr, Cr←0, C[32:13]←PIR, C[12:13]←PBR, C[35:3]←PSR, PSR←X[35:3], PBR←X[13:14] PIR←X[32:13], ADJ(0,0), Br←Ar, Ar←Cr, SBR←[S]+1, SBR←SBR-F, F←SBR+F, B[13:14]←SBR[13:14], ADJ(0,0), Display update if needed, LL←X[18:5], Cr←M@PBR, PROF←0. Sequence error Interrupt←C[50:3] ≠ 011 Presence bit Interrupt←CR47 ≠ 1 PBR←C[19:20], Pr←PIR + PBR, and (NC)	0	0	C5
PAF	STFF	C	Stuff environment	1	2	If A[50:3] ≠ 001; invalid interrupt occurs. If A[50:3] = 001 and AR46 = 1; (NC) If A[50:3] = 001 and AR46 ≠ 1; Dnn←A[13:5], BUF←[Dnn], BUF2←[Dnn], Cr←M@[Dnn], If C[50:3] ≠ 011; sequence error interrupt occurs, If C[50:3] = 011 and CR47 = 1 or CR46 = 0; BUF4←C[13:14], BUF2←[BUF2] - [BUF4], and Cr←M@[BUF2] If C[50:3] = 011, CR47 = 0, and CR46 = 1; A[45:10]←C[45:10], C[29:10]←C[45:10],	1	2	C48

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
PAF (Cont)						If [SNR] = C[39:20]; A[35:16]←[BOSR] + C[35:16], A[18:5]←0, AR46←1, and (DC) If [SNR] ≠ C[39:20]; Cr←M@[D00]+2, presence bit interrupt←CR47 = 0, TEMP←C[39:20], C[39:20]←0, C[29:10]←A[45:10], invalid index interrupt←TEMP>C[39:20], Cr←M@C[19:20] + C[39:20], presence bit interrupt←C47 = 0, A[35:16]←[BOSR]+C[35:16], A[18:5]←0, AR46←1, and (DC)			
VAF	MVST	C	Move to stack	0	1	[Cr]←0, C[13:14]←[S] - [F], C[18:5]←LL, C[35:16]←[S] - [BOSR], C[50:3]←011, M←Cr@[BOSR], BUF2←[D00] + 2, Cr←M@[D00] + 2, presence bit interrupt occurs if CR47 = 0, pseudo call on P87, invalid index occurs if B mantissa<0 or B[38:39]≥C[39:20], Ar←M@C[19:20] + A[19:20], SNR←B[19:20], BROF←0, BOSR←A[19:20], LOSR←A[19:20] + A[39:20], TEMP = A[19:20],	0	0	C35

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
VAF (Cont)						Cr←M@[BOSR], BOSR←processor I.D., S←C[35:16] + [BOSR], BUF←C[13:14], F←[S] - [BUF], BUF←[S] - [BUF], stack underflow interrupt occurs if BOSR>F, C←M@[BUF]; sequence error interrupt if C[50:3] ≠ 011, if CR46 = 1; BUF←D[LL] - C[13:14] and Cr←M@[BUF] if BOSR≤F. if CR46 = 0; Ar←Cr, Cr←M@[DOO] + 2, display update if needed and (QC) if CR46 = 1; display update if needed and (QC)			
PB0	ZERO	D	Lit call zero	0	2	[Ar]←0 and (QC)	1	2	D11
PB1	ONE	D	Lit call one	0	2	[Ar]←0, AR00←1, and (QC)	1	2	D12
PB2	LT8	D	Lit call eight bits	0	2	[Ar]←0, [second syllable]←C[7:8], Cr←Ar, and (QC)	1	2	D13
PB3	LT16	D	Lit call sixteen bits	0	2	[Ar]←0, [second syllable]←C[7:8], C[15:8]←C[7:8], [chird syllable]←C[7:8], Ar←Cr, and (QC)	1	2	D14

PRINTED IN U.S. AMERICA
5000276

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
PB4	PUSH	D	Push down stack register	0	0	(DC)	0	0	D1
VB4	STAG	D	Set tag field	1	1	B[50:3]←A[2:3]	0	1	D18
PB5	DLET	D	Delete top of stack	1	2	AROF←0	0	2	D2
VB5	RTAG	D	Read tag field	1	2	C[2:3]←A[2:3], Ar←0, and Ar←Cr	1	2	D17
PB6	EXCH	D	Exchange	1	1	Ar↔Br	1	1	D7
VB6	RSUP	D	Rotate stack up	1	1	Cr←Ar, AROF←0, ADJ(1,1), Ar↔Br, ADJ(1,0), Br←Cr,	1	1	D6
PB7	DUPL	D	Duplicate top of stack	0	1	Ar←Br and Xr←Yr	1	1	D3
VB7	RSDN	D	Rotate stack down	1	1	Cr←Br, ADJ(1,1), Ar↔Br, ADJ(1,0), Br←Cr, and Ar↔Br	1	1	D5
PB8	STOD	D	Store destructive	1	1	A←Br@Ar, memory protect interrupt if M@Ar has MB48 = 1	0	0	D30

PRINTED IN U.S. AMERICA
5000276

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
VBB	RPRR	D	Read processor register	1	2	Cr←Ar, A[19:20]←ICM@C[5:6]	1	2	D19
PB9	STON	D	Store non-destructive	1	1	M←Br@[Ar], memory protect interrupt if M@Ar has MB48 = 1	0	1	D30
VB9	SPRR	D	Set processor register	1	1	ICM←A[19:20]@B[5:6]	0	0	D20
PBA	OVRD	D	Overwrite destructive	1	1	M←Br@[Ar] regardless of MB48	0	0	D30
VBA	RDLK	D	Read with lock	1	1	M←Br@[Ar] and Ar←M@[Ar] regardless of MB48	1	0	D30
PB8	OVRN	D	Overwrite non-destructive	1	1	M←Br@[Ar] regardless of MB48	0	1	D30
VBB	CBON	D	count binary	1	2	Ar←"1"s" in Ar if A[50:3] = 000 Ar←"1"s" in Ar and Xr if A[50:3] = 010	1	2	D21
VBC	LODT	D	Load transparent	1	2	Cr←Ar, Ar←M@[Cr]	1	2	D8
PBD	LOAD	D	Load	1	2	If A[50:3] ≠ 101 or 001; invalid operator interrupt occurs If A[50:3] = 101 and AR45 = 0; invalid operator interrupt occurs			

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
PBD (Cont)						If A[50:3] = 101 and AR47 = 0; presence bit interrupt occurs Cr←Ar, Ar←M@[Cr], If A[50:3] = 000, 100, 110, or 001; (OC) If A[50:3] = 011 or 111; invalid operator interrupt occurs If A[50:3] = 010; Xr←M@[Ar] and (OC) If A[50:3] = 101, AR47 = 0, and AR46 = 0; A[19:20]←BUF2 and (OC)			
VBD	LLLU	D	Linked list lookup	0	1	If B[50:3] ≠ 0x0; invalid operator interrupt occurs. If B[50:3] = 0x0; Br is integerized and rounded and ADJ(1,1) If B[50:3] ≠ 101, BR47 ≠ 1, or BR45 ≠ 0; invalid operator interrupt occurs. TEMP←B[19:20], BUF←B[39:20] Cr←Ar, Ar←0 and ADJ(0,1) Ar←integerized value of Br, Xr←0, Yr←0, BUF2←C[19:20], BUF3←C[19:20], Y[19:20]←C[19:20], X[19:20]←C[19:20], If BUF2>BUF; invalid index interrupt occurs.	1	0	026

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
VBD (Cont)						If BUF2≤BUF; Cr←M@BUF3 + TEMP, B[27:28]←C[47:28], If C[19:20] = 0; Ar←0, Xr←0, AR00←1, AR46←1, and (OK) If C[19:20] ≠ 0; Ar←Xr, Xr←0 and (OK) If A[27:28]≤C[47:28] X[19:20]←Y[19:20], Y[19:20]←C[19:20], BUF2←C[19:20], BUF3←C[19:20], Y[19:20]←C[19:20], X[19:20]←C[19:20], and the operation is repeated by comparing BUF2 to BUF.			
PBE	LT48	D	Lit call 48 bits	0	2	PSR←0, Ar←0, PROF←0, Pr←M@PIR + PBR, Ar←Pr, and A[50:3]←0	1	2	015
VBE	SRCH	D	Masked search for equal	1	1	If A[50:3] ≠ 101; invalid operator interrupt occurs.	1	0	023
				0	1	If A[50:3] = 101; Cr←Ar, Xr←Br, ADJ(0,1), If C45 = 1; TEMP←C[39:20] and BUF←C[39:20] If C45 = 0; TEMP←C[39:20] - 1 and BUF←C[39:20] - 1 Ar M@ BUF + C[19:20], C[39:20]←BUF, Ar←Ar(LAND)Xr, Br←Br(LAND)Xr, If A[50:3] = B[50:3] and A[47:48] = B[47:48]; Ar←0, Xr←0, A[19:20]←C[39:20], and (OK)			

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
V8E (Cont)						if A[50:3] ≠ B[50:3] or A[47:48] ≠ B[47:48]: TEMP←C[39:30]-1, BUF←C[39:20]-1, and operation returns to Ar←Ar(LAND)Xr, Br←Br(LAND)Xr. If Ar = Br when C[39:20] = 0; AR00←1, AR46←1, and (DC)			
P8F	MPCW	D	Make program control word	0	2	Pr←MSPIR + PBR, Ar←Pr, A[50:3]←111	1	2	D16
PC0	SCLF	E	Scale left	0	1	[Second syllable]::n, Ar←integerized value of Br X 10^n and (DC)	1	0	E1
						If n>12; OFFF←1 and (DC)	0	1	
						If n = 0; (DC)	0	1	
						If Br cannot be integerized; (DC)	0	1	
PC1	DSLF	E	Dynamic scale left	1	1	Integerized value of Ar::n, Ar←integerized value of Br X 10^n, (DC)	1	0	E1
						If n>12; OFFF←1 and (DC)	1	1	
						If n = 0; (DC)	1	1	
						If Br cannot be integerized; (DC)			
PC2	SCRT	E	Scale right truncate	0	1	[Second syllable]::n, Ar←integerized value Br/10^n and (DC)	1	0	E1

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
PC2 (Cont)						If n>12; Invalid operator interrupt occurs.	0	1	
						If n = 0; [Br + Yr]←integerized value of Br, B[50:3]←010, and (OC) if Br cannot be integerized; (OC)	0	1	
PC3	DSRT	E	Dynamic Scale right truncate	1	1	Integerized value of Ar::n, Ar←integerized value of Br/10^n and (OC) If n 12; Invalid operator interrupt occurs.	1	0	E1
						If n = 0; [Br + Yr]←integerized value of Br, B[50:3]←010, and (OC)	0	1	
						If Br cannot be integerized; (OC)	0	1	
PC4	SCRS	E	Scale right save	0	1	[Second syllable]::n, Ar←integerized value of Br/10^n, Br←remainder of Br/10^n, and (OC) If n>12; Invalid operator interrupt occurs.	1	1	
						If n = 0; [Br +Yr]←integerized value of Br, B[50:3] 010, and (OC)	0	1	
						If Br cannot be integerized; (OC)	0	1	
PC5	DSRS	E	Dynamic scale right save	1	1	Integerized value of Ar::n, Ar←integerized value of Br/10^n, Br←remainder of Br/10^n, and (OC) If n>12; Invalid operator interrupt occurs.	1	1	E1

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
PC5 (Cont)						If n = 0; [Br + Yr]←integerized value of Br, B[50:3]←010, and (QC) If Br cannot be integerized; (QC)	0 0	1 1	
PC6	SCRF	E	Scale right final	0	1	[Second syllable]::n, Br←remainder of $Br/10^n$ and (QC) If n>12; Invalid interrupt occurs. If n = 0; [Br + Yr]←integerized value of Br, B[50:3]←010, and (QC) If Br cannot be integerized; (QC)	0	1	E1
PC7	DSRF	E	Dynamic scale right final	1	1	Integerized value of Ar::n, Br←remainder of $Br/10^n$ and (QC) If n>12; Invalid interrupt occurs If n = 0; [Br + Yr]←integerized value of Br, B[50:3]←010, and (QC) If Br cannot be integerized; (QC)	0	1	E1
PC8	SCRR	E	Scale right round	0	1	[Second syllable]::n, Ar←integerized value $Br/10^n$, Ar←Ar + 1 if the MSD of the remainder of $Br/10^n \geq 4$. If n>12; Invalid interrupt occurs. If n = 0, [Br + Yr]←integerized value of Br, B[50:3]←010, and (QC) If Br cannot be integerized; (QC)	1 0 0	0 1 1	

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
PC9	DSRR	E	Dynamic scale right round	1	1	Integerized value of Ar::n, Ar←integerized value of Br/10^n, Ar←Ar + 1 if the MSD of the remainder of Br/10^nx4. If n>12; invalid operator interrupt occurs. If n = 0; [Br + Yr]←integerized value of Br, B[50:3]←010, and (0C) If Br cannot be integerized; (0C)	1 0 0	0 1 1	
PCA	ICVD	E	Input convert, destructive	1	1	Integerized value of Ar:: length of decimal digits. If B[50:3] = 0x0; [Br] = decimal digits. If B[50:3] = 101; [Br]:: location of decimal digits, if B[4:5]>23; invalid operator interrupt occurs. (PS)←PDI, Yr←Br, Br←Yr, CNTR←14 If MSD[Yr]≤7; BB[3:4]←MSD[Yr] If MSD[Yr] = 8; BB[3:4]←6, CRIN←1, AA00←1 If MSD [Yr] = 9; BB[3:4]←7, CRIN←1, AA00←1 AA←CC<<1 bit, BB←CC<<1 octade, CNTR←CNTR-1] repeated until CNTR = 0 If Y[50:3] = 000; Yr←CC[48:9], Br←Yr, Yr←CC[39:40], If Br = 0; Br←Yr, B[50:3]←000 and (0C) If Br ≠ 0; B[50:3]←010 and (0C)	0	1	EID

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
PCA (Cont)						If Y[50:3] = 010; Yr←Br, Br←Yr, CNTR←14, repeat MSD[Yr]←adder above, Cr←CC, MSD[Yr]←adder above, Br←CC, Yr←Cr, B[50:3]←010 and (QC)			
PCB	ICVU	E	Input convert, update	1	1	Performs same operation as ICVD (PCA) with the exception that the A register contains the updated source pointer	1	1	E10
PCC	SNGT	E	Set to single precision, truncated	0	1	If B[50:3] = 101; B[39:20]←B[39:20]x2, BR40←0, and (QC) If BR40 = 1 and BR45 = 0 If B[50:3] ≠ 101; Br and Yr are normalized. If B[50:3] = 000; Yr←0 and (QC) If B[50:3] = 010; Yr←0 and (QC) If B[50:3] ≠ 0x0 or 101; invalid operator interrupt occurs.	0	1	E16
PCD	SNGL	E	Set to single precision, rounded	0	1	Br and Yr are normalized If B[50:3] = 000; Yr←0 and (QC) If B[50:3] = 010; Br←Br + Y[38:1], B[50:3]←000, Yr←0 and (QC) If B[50:3] ≠ 0x0; Invalid operator interrupt occurs	0	1	E16

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
PCE	XTND	E	Set to double precision	0	1	If B[50:3] = 101; B[39:20]←B[39:20]/2, BR40←1, and (DC) if BR40 = 0 and BR45 = 0. If B[50:3] = 010; (QE) If B[50:3] = 000; Yr←0, B[50:3]←010, and (IK) If B[50:3] ≠ 0X0 or 101; Invalid operator interrupt occurs	0	1	E16
PCF	IMKS	E	Insert mark stack	1	1	Cr←Br, Br←D, B[50:3]←011, TEMP←S+1, TEMP←TEMP-F, F←TEMP+F, B[13:14]←TEMP, ADJ(1,0), Br←Cr, and BROF←1	1	1	E14
EDO	MINS	F	Move with insert	-	-	If EXSF = 0; [Second syllable]:: length of character to be moved, [Third syllable]:: character to be inserted, C[7:8]←[Second syllable] If EXSF = 1; [Cr]:: length of characters to be moved [Second syllable]:: character to be inserted	-	-	F16

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
E00 (Cont)						RPF←[Cr], Br←M@DBR+DIR, Ar←M@SBR+SIR, LHFF←1 if length = 0 if SSZ = 0*DSZ = 0; SSZ←8 and DSZ←8 if SSZ = 0*DSZ ≠ 0; SSZ←DSZ if SSZ ≠ 0*DSZ = 0; DSZ←SSZ {if FLTF = 0*SCN = 0; (B[character]::DIB)←[Third syllable] and RPF←RPF-1} {if FLTF = 0*SCN ≠ 0; (B[character]::DIB)←(A[character]::SIB), RPF←RPF-1, and FLTF←1} {if FLTF = 1; (B[character]::DIB)←(A[character]::SIB), and RPF←RPF-1} until RPF = 0; LHFF←1 if LHFF = 1; M←Br@DBR+DIR, DIR←DIR+1, and (OP) if EXSF = 1; (Pst)←ENDE			
PD0	TEE0	F	Table enter edit destructive	0	1	if B[50:3] ≠ 101; Invalid operator interrupt occurs. if BR47 = 0; presence bit interrupt occurs			F4

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
PDD (Cont)						TBR←B[19:20], TIR←B[35:16], Cr←Br, Yr←Br, BROF←0 If BR44 = 1; RNTF←1, ADJ[0,1], Yr←Cr, Cr←Br, BROF←0, and continue as if BR44 = 0			
						[BR44 = 0; ADJ[1,1] If A[50:3] = 0X0; DBR←B[19:20], DSZ←B[42:3], DIB←B[39:4], DIR←B[35:16], PSR←Y[38:3] and (DC) If A[50:3] = 101; SBR←A[19:20], SSZ←A[42:3], SIB←A[39:4], SIR←A[35:16], DBR←B[19:20], DSZ←B[42:3], DIB←B[39:4], DIR←B[35:16], PSR←Y[38:3] and (DC)	1 0	0 0	
VDD	USND	F	Unpack signed, destructive	1	1	If B[50:3] ≠ 0X0; Invalid operator interrupt occurs, (PSD)←EXSD, If C[19:20]≥25; Invalid operator interrupt occurs, RPF←C[19:20] SSZ←4. If RPF = 0; LHFF←1, (PSD)←ENDE, and (DC). If RPF ≠ 0: Br←M@DBR←DIR and LHFF←0. If EXTF = 1; [last digit]←1101 if DSZ = 4 [last digit zone]←10 if DSZ = 6 [last digit zone]←1101 if DSZ = 8			F30

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
VDO (Cont)						If EXTF = 0; [1st digit]←1100 if DSZ = 4 [last digit zone]←00 if DSZ = 6 [last digit zone]←1111 if DSZ = 8 {If LHFF = 0*DGSF = 0; (B[character]::DIB)←(A[character]::SIB), RPF←RPF-1 SIB←SIB+1 and DIB←DIB+1} until DGSF = 1 or LHFF = 1 If DGSF = 1; M←Br@DBR+DIR; DIR←DIR+1, DGSF←0, and Br←M@DBR+DIR. If LHFF = 1; (Ps)←ENDE and (OC)	0	1	
ED1	MFLT	F	Move with float	-	-	If EXSF = 0; [Second syllable]:: length of character to be moved [Third syllable]:: character to be inserted, C[7:8]←[Third syllable] [Fourth syllable]:: character to be inserted [Fifth syllable]:: character to be inserted	-	-	F16

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
ED1 (Cont)						If EXSF = 1; [Cr]:: length of character to be moved [Second syllable]: character to be inserted [Third syllable]:: character to be inserted [Fourth syllable]:: character to be inserted RPF←[Cr], Br←M@DBR+DIR, Ar←M@SBR+SIR. LHFF←1 if length = 0. {if FLTF = 0^SCN = 0; {B[character]::DIB)←[Third syllable] and RPF←RPF+1 until SCN ≠ 0} {if FLTF= 0^SCN ≠ 0^EXTF = 1; (B[character]::DIB)←[Fourth syllable], RPF←RPF+1, and FLTF←1} {if FLTF= 0^SCN≠0^EXTF = 0; (B[character]::DIB)←[Fifth syllable], RPF←RPF+1, and FLTF←1}			

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
ED1 (Cont)						(If FLTF = 1; (B[character]::DIB)←(A[character]::SIB) and RPF←RPF-1 until RPF = 0; LHFF←1) If LHFF = 1; M←BR@DBR+DIR, DIR←DIR+1, and (OC)			
PD1	PACD	F	Pack destructive	-	-	[Ar]:: length and [Br]:: source If [Ar]≥25; Invalid operator interrupt occurs. (PsC)←EXPD, BUF←A[19:20], RPF←A[19:20], (If B[50:3] = 101; ADJ(0,0), Ar←M@SBR+SIR, (B[character]::DIB)←A[character]::SIB and RPF←RPF-1) until RPF = 0; ADJ(0,2), (PsC)←ENDE, and (OC)	-	-	F35

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
VD1	UABD	F	Unpack absolute, destructive	1	1	If B[50:3] ≠ 0x0; invalid operator interrupt occurs, (PSC)→EXSD; if C[19:20] ≥ 25; invalid operator interrupt occurs, RPF←C[19:20], SSZ←4. If RPF = 0; (PSC)→ENDE and (QC). If RPF ≠ 0;←M@DBR + DIR, (B[character]::DIC)←(A[character]::SIB), and RPF←RPF-1.	0	0	F30
						{If DGSF = 1 * LHFF = 0; M←Br@DBR + DIR, DIR←DIR+1, Br←M@DBR + DIR, (B[character]::DIB)←(A[character]::SIB), and RPF←RPF-1}	0	0	
						{If LHFF = 1:(PSC)→ENDE and (QC)	0	1	
ED2	SFSC	F	Skip forward source character	-	-	If EXSF = 1; [Cr]:: length If EXSF = 0; [second syllable]:: length and Cr←[second syllable] RPF←Cr, BUF←Cr; Ar←M@SBR + SIR, SIR←SIX+1 Ar←M@SBR + SIR, RPF←RPF-[48/552-(SIB+1)]], RPZF←1 if SM20=0, and LHFF←1 if SMEZ=1.} {If RPZF = 0 * LHFF=0; SIR←SIR+1, SIB←0, Ar←M@SBR+SIR, RPF←RPF, BUF←RPF, RPF←RPF-[48/552-(SIB+1)]] until RPZF = 1 or LHFF = 1	-	-	F13

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
ED2 (Cont)						If RPZF=1 or LHFF=1; SM[3:4]+BUF + S1B, S1B+SM[3:4], S1R-S1R-1, and (OC). If EXSF=1; (PSC)+ENDE]			
PD2	EXSD	F	Execute single micro, destructive	0	1	[Br] = length, (PSC)+NTGR, Cr+Br, ADJ(1,1) If A[50:3] = 0x0; SOPF+1 If A[50:3] = 101; SSZ+A[42:3], SBR+A[19:20], S1R+A[35:16], S1B+A[39:4] If B[50:3] ≠ 101 or BR47+0; Invalid operator interrupt occurs If B[50:3] = 101 and BR47=1; DSZ+B[42:3], DBR+B[19:20], D1R+B[35:16], D1B+B[39:4] and (OC)	-	-	F4
VD2	TWFD	F	Transfer while false destructive	1	1	If A[50:3] ≠ 101 or AR47 = 0; Invalid operator interrupt occurs TBR+A[19:20], Cr+Ar, (PSC)+EXSL, Br+Yr, BROF+1, and ADJ (2,0). Br+M@DBR+D1R, RPF+C[19:20], Ar+M@SBR+S1R, (A[character]::S1B) ▶Cr, S1B+S1B+1 {If SSZ = 4; Yr+M@TBR and D1S+ΔC[3:4] If SSZ = 6; Yr+M@TBR + CR05 and D1S+ΔC[4:5] If SSZ = 8; Yr+M@TBR + C[7:3] and D1S-ΔC[4:5] (A[character]::S1B) ▶Cr, RPF+RPF-1, Yr▶Yr::D1S.			F36

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
VD2 (Cont)						if YROO = 1; TFFF←0, SIB←SIB-1, DIB←DIB-1, BROF←1, and (DC) if YROO = 0; (B[character]::DIB)←(A[character]::SIB), SIB←SIB-1 until RPF = 0 if RPF = 0; LHFF←1, BROF←1, and (DC)	0 0	1 1	
ED3	SRSC	F	Skip reverse source characters	-	-	if EXSF = 1; [Cr]:: length if EXSF = 0; [second syllable]:: length and Cr←[second syllable] RPF Cr, BUF←Cr, Ar←M@ SBR + SIR, SIR←SIR-1 Ar←M@ SBR + SIR, RPF←RPF-SIB, RPZF←1 if SM20=0, and LHFF←1 if SMEZ = 1. {if RPZF=0 * LHFF=0; SIR←SIR-1, SIB←ASIB, Ar←M@SBR + SIR, RPF←RPF, BUF←RPF, RPF←RPF-SIB} until RPZF=1 or LHFF=1. if RPZF=1 or LHFF=1; SM[3:4] = BUF-SIB, SIB←SM[3:4], SIR←SIR+1, and (DC) if EXSF=1; (PS)←ENDE.	-	-	F13
PD3	TWSD	F	Transfer words, destructive	-	-	(PS)←EXSD, if SIB=0; SIR←SIR+0 if SIB≠0; SIR←SIR+1 and SIB←0 if DIB=0; DIR←DIR+0 if DIB≠0; DIR←DIR+1 and DIB←0	-	-	F27

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
PD3 (Cont)						(RPF←C[19:20], RPF←RPF-1, Ar←M@SBR+SIR, M←Ar@DBR+DIR, SIR←SIR+1, DIR←DIR+1) until RPF=0; ENDE←1 and (OC) If MB48 = 1; memory protect interrupt occurs			
FD3	TWTD	F	Transfer true destructive	1	1	If A[50:3]≠101 or AR47=0; invalid operator interrupt occurs. TBR←A[19:20], Cr←Ar, (PSC)←EXSD, Br←Yr, BROF←1, and ADJ(0,2). Br←M@DBR+DIR, RPF←C[19:20], Ar←M@SBR+SIR, (A[character]::SIB) ⊕ Cr, SIB←SIB+1 (If SSZ=4; Yr←M@TBR and DIS←ΔC[3:4] If SSZ=6; Yr←M@TBR + CR05 and DIS←ΔC[4:5] If SSZ=8; Yr←M@TBR+C[7:8] and DIS←ΔC[4:5] (A[character]::SIB)⊕Cr, RPF←RPF-1, Yr⊕Yr::DIS. If YR00=0; TFF←0, SIB←SIB-1, DIB←DIB-1, BROF←1 and (OC) If YR00=1; (B[character]::DIB)←(A[character]::SIB), SIB←SIB+1) until RPF = 0 If RPF=0; LHFF←1, BROF←1, and (OC)	0 0	1 1	F36
ED4	RSTF	F	Reset float	-	-	FLTF←0, and (OC) if EXSF=1; (PSC)←ENDE	-	-	F20

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
PD4	TWOD	F	Transfer word, overwrite destructive	-	-	(PSC) →EXSD, If SIB=0; SIR←SIR+0 If SIB≠0; SIR←SIR+1 and SIB←0 If DIB=0; DIR←DIR+0 If DIB≠0; DIR←DIR+1 and DIB←0 {RPF←C[19:20], RPF←RPF-1, Ar←M@SBR+SIR, M←Ar@DBR+DIR, SIR←SIR+1, DIR←DIR+1} until RPF=0; ENDE←1 and (QC)	-	-	F27
VD4	SWFD	F	Scan while false, destructive	1	1	If A[50:3]≠101 or AR47=0; invalid operator interrupt occurs. TBR←A[19:20], Cr←Ar, (PSC) →EXSD, Sr←Yr, BROF←1, and ADJ←(0,2) RPF←C[19:20], Ar←M@SBR+SIR, (A[character]::SIB)⇒Cr, SIB←SIB+1. {If SSZ=4; Yr←M@TBR and DIS←ΔC[3:4] If SSZ=6; Yr←M@TBR+CRDS and DIS←ΔC[4:5] If SSZ=8; Yr←M@TBR+C[7:3] and DIS←ΔC[4:5] (A[character]::SIB)⇒Cr, RPF←RPF-1. Yr⇒Yr::DIS. If YROO=1; TFFF←0, SIB←SIB-1, and (QC) If YROO=0; SIB←SIB+1} until RPF=0 If RPF=0; LHFF←1, and (QC)	-	-	F36

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
ED5	ENDF	F	End float	-	-	[second syllable] = character [third syllable] = character Br+M@DBR+DIR, C[7:8]+[second syllable] If FLTF=0 * EXTF=0; C[7:8]+[third syllable], DIB+DIB+1, (B[character]::DIB)+C[7:8], M+Br@DBR+DIR DIR+DIR+1, BROF+0 and (DC) If EXSF=1; (PsC)+ENDE If FLTF=0 * EXTF=1; CPSR+1, DIB+DIB+1, (B[character]::DIB)+C[7:8]; M+Br@DBR+DIR, DIR+DIR+1, BROF+0 and (DC) if EXSF=1; (PsC)+ENDE If FLTF=1; FLTF+0 and (DC) If EXSF=1; (PsC)+ENDE	-	-	F9
PD5	SISO	F	String isolate	-	-	(PsC)+EXPL, If SSZ=4; C[19:20]-24 If SSZ=6; C[19:20]-16 If SSZ=8; C[19:20]-12 If SM2D=1; Invalid operator Interrupt occurs If SMEZ=1; Br+0, Yr+0 and (DC) If SM2D=0 * SMEZ=0; DIB+(48/SSZ-12) if SSZ=4 DIB+(48/SSZ-8) if SSZ=6 DIB+(48/SSZ-6) if SSZ=8	0	1	F24

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
PD5 (Cont)						If SM20/=1; DIB←(48/SSZΔ)48/SSZ-BUF and B[50:3]←010 SIB(B≡) - DIB(B≡) If SM20/=1; TOA←DIB(B≡), DIS←[SIB(B≡) - DIB(B≡)]-16, TOM←Δ[SIB(B≡)-DIB(B≡)], If SM20/=0; TOA←DIB(B≡), TOM←47, and DIS←SIB(B≡) - DIB (B) Br←Ar::TOA,TOM, and DIS If B[50:3]=000 and (QC) Br←Ar::TOA, TOM, and DIS, and Yr←Xr::TOA,TOM, and DIS if B[50:3]=000 and (QC)			
VD5	SWTD	F	Scan while true, destructive	1	1	If A[50:3]≠101 or AR47=0; invalid operator interrupt occurs TBR←A[19:20], Cr←Ar, (PSC)←EXSD, Br←Yr, BROF←1, and ADJ(0,2) RPF←C[19:20], Ar←MOSBR+SIR, (A[character]::SIB▷Cr, SIB←SIB+1, {If SSZ=4; Yr←MOTBR and DIS←ΔC[3:4] If SSZ=6; Yr←MOTBR + CR05 and DIS←ΔC[4:5] If SSZ=8; Yr←MOTBR + C[7:3] and DIS←ΔC[4:5] (A[character]::SIB▷Cr, RPF←RPF+1, Yr▷Yr::DIS. If YR00=0; TFFF←0, SIB←SIB-1, and (QC) If YR00=1; SIB←SIB+1} until RPF=0 If RPF=0; LHFF←1, and (QC)	-	-	F36

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
ED6	MVNU	F	Move numeric unconditional	-	-	If EXSF=1; [Cr] = length If EXSF=0; [second syllable] = length and C[7:8]←[second syllable] RPF←C[7:8], Br←M@DBR+DIR, Ar←M@SBR+SIR If SSZ=0 * DSZ=0; SSZ←8 and DSZ←8 If SSZ=0 * DSZ≠0; SSZ←DSZ If SSZ≠0 * DSZ=0; DSZ←SSZ ((A[character]::S18)→(B[character]::DIB), DIB←DIB+1, SIB←SIB+1, and RPF←RPF-1) until RPF=0; LHFF←1 if LHFF=1; M←Br@DBR+DIR, DIR←DIR+1 and (DC) if EXSF=1; (PSC)←ENDE	-	-	F16
ED6	SXSN	F	Set external sign	-	-	If AROF=1; EXTF←AR46 and (QC) If AROF=0; ADJ(0,1), EXTF←BR46 and (DC)	1 0	0 1	F42
ED7	MCHR	F	Move characters	-	-	If EXSF=1; Cr=length If EXSF=0; [second syllable] = length and C[7:8]←[second syllable] ((A[character]::S18)→(B[character]::DIB), DIB←DIB+1, SIB←SIB+1, and RPF←RPF-1) until RPF=0; LHFF←1 if LHFF=1; M←Br@DBR+DIR, DIR←DIR+1 and (DC) if EXSF=1; (PSC)←ENDE	-	-	F16
ED7	ROFF	F	Read and clear overflow flip-flop	-	-	If AROF=1; ADJ(0,2), If AROF=0; Ar←0 AR00←OFFF, DFFF←0, and (DC)	0	2	F43

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
VD7	TRNS	F	Translate	1	1	If A[50:3]≠101 or AR4]≠0; Invalid operator interrupt occurs TBR←A[19:20], Cr←Ar, (Ps)C←EXSL, Br←Yr, BROF+1; and ADJ(0,2) Br←M@DBR+DIR, RPF←C[19:20], Ar←M@SBR+SIR, (A[character]::SIB) ⊳Cr, SIB←SIB+1 (If SSZ=4; Yr←M@TBR + C[3:2] and DIS[4:2]←C[1:2] If SSZ=6; Yr←M@TBR + C[5:4] and DIS[4:2]←C[1:2] If SSZ=8; Yr←M@TBR + C[7:6] and DIS[4:2]←C[1:2] (A[character]::SIB)⊳Cr, RPF←RPF-1, Yr⊳Yr::DIS, DIS←DFRJ, (B[character]::DIB)←(Y[character]::DIS), SIB←SIB+1, DIB←DIB+1) until RPF=0 IF RPF=0; LHFF+1 and (DC)	-	-	F36
EDB	INOP	F	Insert overpunch	-	-	Br←M@DBR+DIR, If DSZ=4; Invalid operator interrupt occurs, If DSZ=8;C[7:4]←1101 If DSZ=6;C[5:2]←10 DIB←DIB+1, TOA←DIB(B=), TOM←DIB(B=), TOM←TOM-4, If DSZ=8; (B[zone]::DIB)←C[7:4], M←Br@DBR+DIR, DIR←DIR+1 and (DC) If DSZ=6;(B[zone]::DIB)←C[5:2]), M←Br@DBR+DIR, DIR←DIR+1, and (DC)	-	-	F9

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
PDB	TEEU	F	Table enter edit, update	0	1	If B[50:3]≠101 or BR47=0; invalid operator interrupt occurs TBR←B[19:20], TIR←B[35:16], Cr←Br, Yr←Br, BROF←0, if BR44=1; RNTF←1, ADJ(0,1), Yr←Cr, Cr←Br, BROF←0, and continue as if BR44=0 If BR44=0; ADJ(1,1) If A[50:3] = 0x0; DBR←B[19:20], DSZ←B[42:3], DIB←B[39:4], DIR←B[35:16], PSR←Y[38:3], UPDF←1, and (QC) if A[50:3]=101; SBR←A[19:20], SSZ←A[42:3], SIB←A[39:4], SIR←A[35:16], DBR←B[19:20], DSZ←B[42:3], DIB←B[39:4], DIR←B[35:16], PSR←B[38:3], UPDF←1, and (QC)	1 0	0 0	F4
VD8	USNU	F	Unpack signed update	1	1	If B[50:3]≠0x0; Invalid operator interrupt occurs (Ps)←EXSU, If C[19:20] ≥ 25; invalid operator interrupt occurs RPF←C[19:20], SSZ←4, if RPF=0; LHFF←1, (Ps)←ENDE and (QC) if RPF≠0; Br←M@DBR+DIR and LHFF←0 If EXTF=1; ([1st digit]←1101 if DSZ=4 [last digit zone]←10 If DSZ=6 [last digit zone]←1101 if DSZ=8	0	0	F30

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
VD8 (Cont)						If EXTF=0; [1st digit]←1100 if DSZ=4 [last digit zone]←00 if DSZ=6 [last digit zone]←1111 if DSZ=8 (If LHFF=0 * DGSF=0; (B[character]::DIB)←(A[character]::SIB), RPF←RPF-1, SIB←SIB+1, and DIB←DIB+1) until DGSF=1 or LHFF=1 If DGSF=1; M←Br@DBR+DIR, DIR←DIR+1, DGSF←0 and Br←M@DBR+DIR If LHFF=1; (PSE)←ENDE and (OC)	0	1	
E09	INSG	F	Insert display sign	-	-	[syllable two] = character and [syllable three] = character Br←M@DBR+DIR,C[7:8]←[syllable two] If EXTF=1; Br←C[7:8]::DIB, M←Br@DBR+DIR, DIR←DIR+1, and (OC) If EXTF=0; C[7:8]←[third syllable], Br←C[7:8]::DIB, M←Br@DBR+DIR, DIR←DIR+1, and (OC)	-	-	F9
PD9	PACU	F	Pack update	-	-	[Ar]::length and [Br]::source If [Ar] ≥ 25; invalid operator interrupt occurs (PSE)←EXPU, BUF←A[19:20], RPF←A[19:20] (If B[50:3]=101; ADJ(0,0), Ar←M@SBR+SIR, (B[character]::DIB)←(A[character]::SIB) and RPF←RPF-1)			

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
PD9 (Cont)						until RPF=0; ADJ(2,0), (PSC)=ENDE, and (DC)			
VD9	VABU	F	Unpack absolute, update	1	1	if B[50:3]≠0x0; Invalid operator interrupt occurs (PSC)=EXSU, if C[19:20] ≥ 25; Invalid operator interrupt occurs RPF=C[19:20], SSZ=4. If RPF=0; (PSC)=ENDE and (DC)	0	0	F30
						If RPF≠0; Br=M@DBR+DIR, (B[character]::DIB)=(A[character]::SIB), and RPF=RPF-1			
						{If DGSF=1 * LHFF=0; M=Br@DBR+DIR, DIR=DIR+1, Br=M@DBR+DIR, (B[character]::DIB)=(A[character]::SIB), and RPF=RPF-1}			
						If LHFF=1; (PSC)=ENDE and (DC)	0	1	
EDA	SFDC	F	Skip forward destination character	-	-	If EXSF=1; [Cr]::length If EXSF=0; [second syllable]::length and Cr=[second syllable] RPF=Cr, BUF=Cr, Br=M@DBR+DIR, DIR=DIR+1, Br=M@DBR+DIR, RPF=RPF-[48/DSZ-(DIB+1)], RPZF=1 If SM20=0 and LHFF=1 If SMEZ=1	-	-	F13
						{If RPZF=0 * LHFF=0; DIR=DIR+1, DIB=0, Br=M@DBR+DIR, RPF=RPF-1, BUF=RPF, RPF=RPF-[48/DSZ-(DIB+1)]} until RPZF=1 or LHFF=1			

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
EDA (Cont)						If RP2F=1 or LHFF=1; SM[3:4]+BUF+DIB, DIB+SM[3:4], DIR+DIR-1 and (0C). If EXSF=1; (Ps)>ENDE			
PDA	EXSU	F	Execute single micro, update	0	1	[Br]=length, (Ps)>NTGR, Cr+Br, ADJ(1,1), If A[50:3]=0x0; SOPF+1 If A[50:3]=101; SSZ+A[42:3], SBR+A[19:20], SIR+A[35:16], SIB+A[39:4]. If B[50:3]≠101 or BR47≠0; Invalid operator interrupt occurs If B[50:3]=101 and BR47=1; DSZ+B[42:3], DBR+B[19:20], DIR+B[35:16], DIB+B[39:4], UPDF+and(0C)	-	-	F4
VDA	TWFU	F	Transfer while false, update	1	1	If A[50:3]≠101 or AR47≠0; Invalid operator interrupt occurs TBR+A[19:20], Cr+Ar, (Ps)>EXSL, Br+Yr, BROF+1 and ADJ(0,2). Br+M@DBR+DIR, RPF+C[19:20], Ar+M@SBR+SIR, (A[character]::SIB)>Cr, SIB+SIB+1 {If SSZ=4; Yr+M@TBR and DIS+ΔC[3:4] If SSZ=6; Yr+M@TBR+CR05 and DIS+ΔC[4:5] If SSZ=8; Yr+M@TBR+C[7:3] and DIS+ΔC[4:5] (A[character]::SIB)>Cr, RPF RPF-1, Yr>Yr::DIS			F36

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
VDA (Cont)						If YROO=1; TFFF←0, SIB←SIB-1, DIB←DIB-1, BROF←1, ADJ(0,2), Ar←RPF and (DC)	1	2	
						If YROO=0; (S[character]::DIB)←(A[character]::SIB), SIB←SIB+1) until RPF=0 If RPF=0; BROF←1, ADJ(0,2), Ar←RPF, and (DC)	1	2	
EDB	SRDC	F	Skip reverse, destination characters	-	-	If EXSF=1; [Cr]=length If EXSF=0; [second syllable]=length and Cr←[second syllable] RPF←Cr, BUF←Cr, Br←M@DBR+DIR, DIR←DIR-1, Br←M@DBR+DIR, RPF←RPF-DIB, RPZF←1 If SM20=0, and LHFF←1 if SMEZ=1.	-	-	F13
						{If RPZF=0∧LHFF=0; DIR←DIR-1, DIB←ΔDIB, Br←M@DBR+DIR, RPF←RPF, BUF←RPF, RPF←RPF-DIB} until RPZF=1 or LHFF=1			
						If RPZF=1 or LHFF=1; SM[3:4]=BUF-DIB, DIB←SM[3:4], DIR←DIR+1, and (DC) If EXSF=1; (Ps)←ENDE			
PDB	TWSU	F	Transfer words, update	-	-	(Ps)←EXSU, If SIB=0; SIR←SIR+0 If SIB≠0; SIR←S(R+1 If DIB=0; DIR←DIR+0 If DIB≠0; DIR←DIR+1 {RPF←C[19:20], RPF←RPF-1, Ar←M@SBR+SIR, M←Ar←DBR+DIR, SIR←SIR+1, DIR←DIR+1} until RPF=0; ENDE←1 and (DC) If MB48=1; memory protect interrupt occurs	-	-	F27

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
VDB	TWTU	F	Transfer while true, update	1	1	If A[50:3]≠101 or AR47=0; invalid operator interrupt occurs TBR←A[19:20], Cr←Ar, (PsC)←EXSU, Br←Yr, BROF←1, and ADJ(0,2) Br←M@DBR+DIR, RPF←C[19:20], Ar←M@SBR+SIR, (A[character]::SIB)⇒Cr, SIB←SIB+1 {If SSZ=4; Yr←M@TBR and DIS←ΔC[3:4] If SSZ=6; Yr←M@TBR+CR05 and DIS←ΔC[4:5] If SSZ=8; Yr←M@TBR + C[7:3] and DIS←ΔC[4:5] (A[character]::SIB)⇒Cr, RPF←RPF-1, Yr⇒Yr::DIS. If YR00=0; TFFF←0, SIB←SIB-1, DIB←DIB-1, BROF←1, ADJ(0,2), Ar←RPF and (OC) If YR00=1; (B[character]::DIB)←(A[character]::SIB); SIB←SIB+1} until RPF=0. If RPF=0; BROF←1, ADJ(0,2), Ar←RPF, and (OC)	1	2	F36
EDC	INSU	F	Insert unconditional	-	-	If EXSF=1;[Cr]=length and [second syllable]=character. If EXSF=0; [second syllable]=length, [third syllable]=character, and Cr←[second syllable] RPF←Cr, Br←M@DBR+DIR, Cr←[character], {DIB←DIB+1, RPF←RPF-1, (B[character]::DIB←Cr} until RPF=0 or DIB=0 If DIB=0; M←Br@DBR+DIR, DIR←DIR+1, Br←M@DBR+DIR. If RPF=0; M←Br@DBR+DIR, DIR←DIR+1, and (OC) If EXSF=1; (PsC)←ENDE	1 -	2 -	F11

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
PDC	TWDU	F	Transfer words overwrite, update	-	-	(PsC)+EXSU, if S1B=0; S1R+S1R+0 if S1B≠0; S1R+S1R+1 if D1B=0; D1R+D1R+0 if D1B≠0; D1R+D1R+1 (RPF+C[19:20], RPF+RPF-1, Ar+MESBR+S1R, M+Ar@D0R+D1R, S1R+S1R+1, D1R+D1R+1) until RPF=0; ENDE+1 and (QE)	-	-	F27
VDC	SWFU	F	Scan while false update.	1	1	If A[50:3]≠101 or AR47=0; invalid operator interrupt occurs. TBR+A[19:20], Cr+Ar, (PsC)+EXSU, Br+Yr, BROF+1, and ADJ(0,2) RPF+C[19:20], Ar+MESBR+S1R, (A[character]::S1B)⊳Cr, S1B+S1B+1 {if SSZ=4; Yr+MBTBR and D1S+ΔC[3:4] if SSZ=6; Yr+MBTBR+CRDS and D1S+ΔC[4:5] if SSZ=8; Yr+MBTBR+C[7:3] and D1S+ΔC[4:5] (A[character]::S1B)⊳Cr, RPF+RPF-1, Yr⊳Yr::D1S. If YRDO=1; TFFF+0, S1B+S1B-1, ADJ(0,2), Ar+RPF, and (QE) If YRDO=0; S(B+S1B+1) until RPF=0 If RPF=0; LHFF+1, and (QE)	-	-	F36
EDD	INSC	F	Insert conditional	-	-	If EXSF=1; [Cr] = length, [second syllable]=1st character, and [third syllable]=2nd character	-	-	F33

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
EDD (Cont)						if EXSF=0; [second syllable]=length, [third syllable]=1st character, [fourth syllable]=2nd character, and Cr←[second syllable] RPF+Cr, Br←M@DBR+DIR, Cr←[1st character] if FLTF=0, and Cr←[2nd character] if FLTF=1. {DIB+DIB+1, RPF+RPF-1, (B[character]::DIB←Cr) until RPF=0 or DIB=0 if DIB=0; M←Br@DBR+DIR, DIR←DIR+1, Br←M@DBR+DIR if RPF=0; M←Br@DBR+DIR, DIR←DIR+1 and (OC) if EXSF=1; (PsC)←ENDE			
PDD	EXPU	F	Execute single micro, single pointer update	0	1	[Br]=length, (PsC)←NTGR, Cr←Br, ADJ(0,1), Ar←Br, AROF+1 if A[50:3]=0x0; SOPF←1, UPDL←1, and (OC) if A[50:3]=101; SSZ←A[42:3], SBR←A[19:20]; SIR←A[35:16], SIB←A[39:4], UPDL←1, and (OC)	-	-	F4
VDD	SWTU	F	Scan while true, update	1	1	if A[50:3]≠101 or AR47=0; invalid operator interrupt occurs TBR←A[19:20], Cr←Ar, (PsC)←EXSU, Br←Yr, BROF←1, and ADJ(0,2) RPF←C[19:20], Ar←M@SBR+SIR, (A[character]::SIB ⊗ Cr, SIB←SIB+1.			F36

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
VDD (Cont)						(If SSZ=4; Yr←M@TBR and DIS←ΔC[3:4] If SSZ=6; Yr←M@TBR+CA05 and DIS←ΔC[4:5] If SSZ=8; Yr←M@TBR+ C[7:3] and DIS←ΔC[4:5] (A[character]::SIB@Cr, RPF←RPF+1, Yr←Yr::DIS. If YROO=0; TFFF←0, SIB←SIB-1,ADJ(0,2),Cr←RPF,and (QC)	1	2	
						If YROO=1; SIB←SIB+1) until RPF=0 If RPF=0; LHFF←1, ADJ(0,2), Cr←RPF, and (QC)	1	2	
EDE	ENDE	F	End edit	-	-	M←Br@DBR+DIR, BR47←1, If UPDF=0; (QC) if UPDF=1; BR47←1, BR46←1, BR43←DPRF, B[39:4]←DIB, BR45←1, B[42:3]←DSZ, B[50:3]←101, B[35:16] ←DIR, B[19:20]←DBR, if A[50:3]= 000; DIS←SIB(8=) If SIB≠0; Cr←Ar:: DIS, Ar←Ar::DIS, and (QC)	0	0	F21
						If A[50:3]=010; DIS←SIB(8=) If SIB≠0, Cr←Ar::DIS, Ar←Xr::DIS, Xr←Cr::DIS and (QC)	1	1	
						If SGPF=0; Cr←Br,BR47←1, BR46←1, BR45←1, B[50:3]←101, B[39:4]←SIB, BR43←SPRF, B[42:3]←SSZ, B[35:16]← DIR, B[19:20]←DBR, Ar←Br, Br←Cr and (QC)	1	1	
							1	1	

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
PDE	RTFF	F	Read true/ false flip-flop	0	2	Ar←0, ARO0←TFFF	1	2	F44
UDF	HALT	F	Conditional halt	-	-	(DC)	-	-	F45
PE0	TLSD	G	Transfer while less, destructive	0	1	Cr←Br, (PsC)←EXSD, Ar←M@SBR+SIR, RPF←Cr, Br←M@DBR+DIR, if (A[character]::SIB≥Y[character]); RPF←RPF-1, M←Br::DBR+DIR, and (PsC)←ENDE if (A[character]::SIB)< Y[character]; RPF←RPF-1, (A[character]::SIB)←(B[character]:: DIB), SIB←SIB+1, DIB←DIB+1) until RPZF=1; M←Br@DBR+DIR, and (PsC)←ENDE	-	-	X1
PE1	TGED	G	Transfer while greater or equal destructive	0	1	Cr←Br, (PsC)←EXSD, Ar←M@SBR+SIR, RPF←Cr, Br←M@DBR+DIR if (A[character]::SIB)<Y[character]; RPF←RPF-1, M←Br::DBR+DIR, and (PsC)←ENDE (if (A[character]::SIB)≥ Y[character]; RPF←RPF-1, (A[character]::SIB)←(B[character:: DIB), SIB←SIB+1, DIB←DIB+1) until RPZF=1; M←Br@DBR+DIR and (PsC)←ENDE	-	-	X1

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
PE2	TGTD	G	Transfer while greater, destructive	0	1	Cr←Br, (PsC)←EXSD, Ar←M@SBR+SIR, RPF←Cr, Br←M@DBR+DIR If (A[character]::SIB≤ Y[character]; RPF←RPF-1, M←Br::DBR+DIR, and (PsC)←ENDE {If (A[character]::SIB)> Y[character]; RPF←RPF-1, (A[character]::SIB)←(B[character]:: DIB), SIB←SIB+1, DIB←DIB+1} until RPZF=1; M←Br@DBR+DIR and (PsC)←ENDE	-	-	X1
PE3	TLED	G	Transfer while less or equal, destructive	0	1	Cr←Br, (PsC)←EXSD, Ar←M@SBR+SIR, RPF←Cr, Br←M@DBR+DIR If (A[character]::SIB>Y[character]; RPF←RPF-1, M←Br@DBR+DIR, and (PsC)←ENDE If (A[character]::SIB)≤ Y[character]; RPF←RPF-1, (A[character]::SIB)←(B[character]:: DIB), SIB←SIB-1, DIB←DIB-1} until RPZF=1; M←Br@DBR+DIR and (PsC)←ENDE	-	-	X1
PE4	TEQD	G	Transfer while equal, destructive	0	1	Cr←Br, (PsC)←EXSD, Ar←M@SBR+SIR, RPF←Cr, Br←M@DBR+DIR If (A[character]::SIB)≠Y[character]; RPF←RPF-1, M←Br@DBR+DIR, and (PsC)←ENDE {If (A[character]::SIB)= Y[character]; RPF←RPF-1, (A[character]::SIB)←(B[character]:: DIB), SIB←SIB+1, DIB←DIB+1} until RPZF=1; M←Br@DBR+DIR and (PsC)←ENDE	-	-	X1

PRINTED IN U.S. AMERICA
5000276

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
PE5	TNED	G	Transfer while not equal, destructive	0	1	Cr←Br, (PsC)←EXSD, Ar←M@SBR+SIR, RPF←Cr, Br←M@DBR+DIR If (A[character]::SIB)= Y[character]; RPF←RPF-1, M←Br@DBR+DIR, and (PsC)←ENDE {If (A[character]::SIB)≠ Y[character]; RPF←RPF-1, (A[character]::SIB)←(B[character]:: DIB), SIB←SIB+1, DIB←DIB+1} until RPZF=1; M←Br@DBR+DIR and (PsC)←ENDE	-	-	X1
PE6	TUND	G	Transfer un-conditional destructive	0	1	Cr←Br, (PsC)←EXSD, Ar←M@SBR+SIR, RPF←Cr, Br←M@DBR+DIR, Yr←0 {RPF←RPF-1, (A[character]::SIB)← (B[character]::DIB), SIB←SIB+1, DIB←DIB+1} until RPZF=1; M←Br@DBR+DIR, and (PsC)←ENDE	-	-	X1
PE8	TLSU	G	Transfer while less, update	0	1	Cr←Br (PsC)←EXSU, Ar←M@SBR+SIR, RPF←Cr, Br←M@DBR+DIR If (A[character]::SIB)≥ Y[character]; RPF←RPF-1, M←Br::DBR+DIR, (PsC)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (PC) {If (A[character]::SIB)< Y[character]; RPF←RPF-1, (A[character]::SIB)←(B[character]:: DIB), SIB←SIB+1, DIB←DIB+1} until RPZF=1; M←Br@DBR+DIR, (PsC)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (PC)	1	1	X1

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
PE9	TGEU	G	Transfer while greater or equal, update	0	1	Cr←Br, PsC→EXSU, Ar←M@SBR+SIR, RPF←Cr, Br←M@DBR+DIR, {If (A[character]::SIB)< Y[character]; RPF←RPF-1, M←Br::DBR+DIR, (PsC)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (DC) {If (A[character]::SIB)≥ Y[character]; RPF←RPF-1, (A[character]::SIB)←(B[character]::DIB), SIB←SIB+1, DIB←DIB+1} until RPZF; M←Br@DBR+DIR, (PsC)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (DC)	1	1	X1
PEA	TGTU	G	Transfer while greater, update	0	1	Cr←Br, (PsC)←EXSU, Ar←M@SBR+SIR, RPF←Cr, Br←M@DBR+DIR {If (A[character]::SIB)≤ Y[character]; RPF←RPF-1, M←Br@DBR+DIR, (PsC)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (DC) {If(A[character]::SIB)> Y[character]; RPF←RPF-1, (A[character]::SIB)←(B[character]::DIB), SIB←SIB+1, DIB←DIB+1} until RPZF=1; M←Br@DBR+DIR, (PsC)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (DC)	1	1	X1

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
PEB	TLEU	G	Transfer while less or equal, update	0	1	Cr←dr, (PSD)←EXSU, Ar←M@SBR+SIR, RPF←Cr, Br←M@DBR+DIR if (A[character]::SIB)> Y[character]; RPF←RPF-1, M←Br@DBR+DIR, (PSD)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (DC) (if (A[character]::SIB)≥ Y[character]; RPF←RPF-1, (A[character]::SIB)←(B[character]:: DIB), SIB←SIB+1, DIB←DIB+1) until RPZF=1; M←Br@DBR+DIR, (PSD)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (DC)	1	1	X1
PEC	TEQU	G	Transfer while equal, update	0	1	Cr←Br, (PSD)←EXSU, Ar←M@SBR+SIR, RPF←Cr, Br←M@DBR+DIR if (A[character]::SIB)≠ Y[character]; RPF←RPF-1, M←Br@DBR+DIR, (PSD)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (DC) (if A[character]::SIB) = Y[character]; RPF←RPF-1, (A[character]::SIB)←(B[character]:: DIB), SIB←SIB+1, DIB←DIB+1) until RPZF=1; M←Br@DBR+DIR, (PSD)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr; and (DC)	1	1	X1

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
PED	TNEU	' G	Transfer while not equal, update	0	1	C←Br, (PsC)←EXSU, Ar←M@SBR+SIR, RPF←Cr, Br←M@DBR+DIR if (A[character]::SIB)≠Y[character]; RPF←RPF-1, M←Br@DBR+DIR, (PsC)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (DC) [(if (A[character]::SIB ≠ Y[character]; RPF←RPF-1, (A[character]::SIB)←(B[character]::DIB), SIB←SIB+1, DIB←DIB+1) until RPZF=1; M←Br@DBR+DIR, (PsC)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (DC)	1	1	X1

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
PEE	TUNU	G	Transfer unconditional, update	0	1	Cr←Br, (PsC)←EXSU, Ar←M@SBR+SIR, RPF←Cr, Br←M@DBR+DIR {RPF←RPF-1, (A[character]::SIB)←(B[character]::DIB), SIB←SIB+1, DIB←DIB+1} until RPZF = 1; M←Br@DBR+DIR, (PsC)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (OK)	1	1	X1
PFO	CLSD	H	Compare characters less, destructive	0	1	Cr←Br, (PsC)←EXSD, Ar←M@SBR+SIR, TFFF←1, RPF←Cr, Br←M@DBR+DIR, {Yr = Br, DIS←SIB(B≡)-DIB(B≡), TOA←DIB(B≡), DIB←DIB+1, TOM←DIB(B≡), RPF←RPF-1. If (A[character]::SIB) ≠ (B[character]::DIB); TFOF←1 If(A[character]::SIB)≥(B[character]::DIB); TFFF←0 DIB←DIB+1, and SIB←SIB+1} until RPZF = 1#SERR = 0 If RPZF = 1 #SERR = 0; TFOF←0, (PsC)←ENDE and (OK)	-	-	X1
VFO	SLSD	H	Scan while less, destructive	0	1	Cr←Br, (PsC)←EXPD, Ar←M@SBR+SIR, RPF←Cr, RPF←RPF-1 If (A[character]::SIB)≥Y[character]; TFFF←0, RPF←RPF+1, DIB←DIB-1, (PsC)←ENDE and (OK)	-	-	X1

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
VF0 (Cont)						{If A[character]::SIB)<Y[character]; SIB←SIB+1, DIB←DIB+1) until RPZF = 1 * SERR = 0 If RPZF = 1 *SERR = 0; (Ps)←ENDE and (0F)			
PF1	CGED	H	Compare characters greater or equal, destructive	0	1	Cr←Br, (Ps)←EXSD, Ar←M@SBR+SIR, TFFF←1, RPF←Cr, Br←M@DBR+DIR {Yr = Br, DIS←SIB(B≡)-DIB(B≡), TOA←DIB(B≡), DIB←DIB+1, TOM←DIB(B≡), RPF←RPF-1 If (A[character]::SIB) ≠ (B[character]::DIB); TFOF←1 If (A[character]::SIB)<(B[character]::DIB); TFFF←0 DIB←DIB+1, SIB←SIB+1) until RPZF = 1 *SERR = 0 If RPZF = 1 *SERR = 0; TFOF←0, (Ps)←ENDE and (0F)	-	-	X1
VF1	SGED	H	Scan while greater or equal, destructive	0	1	Cr←Br, (Ps)←EXPD, Ar←M@SBR+SIR, TFFF←1, RPF←Cr, RPF←RPF-1 If (A[character]::SIB)<Y[character]; TFFF←0, RPF←RPF+1, DIB←DIB-1, (Ps)←ENDE and (0C) {If (A[character::SIB)≥Y[character]; SIB←SIB+1, DIB DIB+1) until RPZF = 1 * SERR = 0; (Ps)←ENDE and (0C)	-	-	X1

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
PF2	CGTD	H	Compare character greater, destructive	0	1	Cr←Br, (Ps)←EXSD, Ar←M@SBR+SIR, TFFF←1, RPF←Cr, Br←M@DBR+DIR (Yr = Br, DIS←SIB(Bπ)-DIB(Bπ), TOA←DIB(Bπ), DIB←DIB+1, TOM←DIB(Bπ), RPF←RPF-1 If (A[character]::SIB ≠ (B[character]::DIB); TFOF←1 If (A[character]::SIB≤(B[character::DIB; TFFF←0, DIB←DIB+1, SIB←SIB+1) until RPZF = 1*SERR = 0 If RPZF = 1*SERR = 0; TFOF←0, (Ps)←ENDE and (OF)	-	-	XI
VF2	SGTD	H	Scan while greater, destructive	0	1	Cr←Br, (Ps)←EXPD, Ar←M@SBR+SIR, TFFF←1, RPF←Cr, RPF←RPF-1 If (A[character]::SIB)≤(B[character]::DIB); TFFF←0, RPF←RPF+1, DIB←DIB-1, (Ps)←ENDE and (OF) (If (A[character]::SIB)>(B[character]::DIB); SIB←SIB+1, DIB←DIB+1) until RPZF = 1*SERR=0; (Ps)←ENDE and (OF)	-	-	XI

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
PF3	CLED	H	Compare characters less or equal, destructive	0	1	Cr←Br, (PsD)←EXSD, Ar←M@SBR+S1R, TFFF←1, RPF←Cr, Br←M@DBR+DIR {Yr = Br, DIS←S1B(B≡)-DIB(B≡), TOA←DIB(B≡), DIB←DIB+1, TOM←DIB(B≡), RPF←RPF-1 if (A[character]::S1B ≠ (B[character]::DIB); TFOF←1 if (A[character]::S1B)>(B[character]::DIB; TFFF←0, DIB←DIB+1, S1B←S1B+1) until RPZF = 1*SERR=0; TFOF←0, (PsD)←ENDE and (DC)	-	-	X1
VF3	SLED	H	Scan while less or equal, destructive	0	1	Cr←Br, (PsD)←EXPD, Ar←M@SBR+S1R, TFFF←1, RPF←Cr, RPF←RPF-1 if (A[character]::S1B)>Y[character]; TFFF←0, RPF←RPF+1, DIB←DIB-1, (PsD)←ENDE and (DC) {if (A[character]::S1B)≤Y[character]; S1B←S(B+1, DIB←DI(B+1)until RPZF = 1*SERR = 0; (PsD)←ENDE and (DC)	-	-	X1
PF4	CEQD	H	Compare character equal, destructive	0	1	Cr←Br, (PsD)←EXSD, Ar←M@SBR+S1R, TFFF←1, RPF←Cr, Br←M@DBR+DIR,	-	-	X1

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
PF4 (Cont)						{Yr = Br, DIS+SIB(B≡)-DIB(B≡), TOA+DIB(B≡), DIB+DIB+1, TOM+DIB(B≡), RPF+RPF-1 If (A[character]::SIB ≠ (B[character]::DIB); TFOF+1 and TFFF+0, DIB+DIB+1, SIB+SIB+1) until RPZF = 1*SERR = 0; TFOF+0, (Ps)+ENDE and (OC)			
VF4	SEQD	. H	Scan while equal, destructive	0	1	Cr+Br, (Ps)+EXPD, Ar+M@SBR+SIR, TFFF+1, RPF+Cr, RPF+RPF-1 If (A[character]::SIB) ≠ Y[character]; TFFF+0, RPF+RPF+1, DIB+DIB-1, (Ps)+ENDE and (OC) {If (A[character]::SIB) = Y[character]; SIB+SIB+1, DIB+DIB+1) until RPZF = 1*SERR = 0; (Ps)+ENDE and (OC)	-	-	X1
PF5	CNED	H	Compare characters not equal, destructive	0	1	Cr+Br, (Ps)+EXSD, Ar+M@SBR+SIR, TFFF+1, RPF+Cr, Br+M@DBR+DIR {Yr = Br, DIS+SIB(B≡)-DIB(B≡), TOA+DIB(B≡), DIB+DIB+1, TOM+DIB(B≡), RPF+RPF-1. If (A[character]::SIB) ≠ (B[character]::DIB); TFOF+1. If (A[character]::SIB) = (B[character]::DIB); TFFF+0, DIB+DIB+1, SIB+SIB+1} until RPZF = 1*SERR=0; TFOF+0, (Ps)+ENDE and (OC)	-	-	X1

PRINTED IN U.S. AMERICA 5000376

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
VF5	SNED	H	Scan while not equal, destructive	0	1	Cr←Br, (PsC)←EXPD, Ar←M@SBR+SIR, TFFF←1, RPF←Cr, RPF←RPF-1 If (A[character]::SIB) = Y[character]; TFFF←0, RPF←RPF+1, DIB←DIB-1, (PsC)←ENDE and (DC) {If (A[character]::SIB) ≠ Y[character]; SIB←SIB+1, DIB←DIB+1} until RPZF = 1*SERR=0; (PsC)←ENDE and (DC)	-	-	X1
PF8	CLSU	H	Compare characters less, update	0	1	Cr←Br, (PsC)←EXSU, Ar←M@SBR+SIR, TFFF←1, RPF←Cr, Br←M@DBR+DIR {Yr = Br, DIS←SIB(B≡)-DIB(B≡), TOA←DIB(B≡), DIB←DIB+1, TOM←DIB(B≡), RPF←RPF-1 If (A[character]::SIB) ≠ (B[character]::DIB; TFOF←1 If (A[character]::SIB)≥(B[character]::DIB; TFFF←0, DIB←DIB+1, and SIB←SIB+1} until RPZF = 1*SERR=0; TFOF←0, (PsC)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr and (DC)	1	1	X1
VF8	SLSU	H	Scan while less, update	0	1	Cr←Br, (PsC)←EXPU, Ar←M@SBR+SIR, RPF←Cr, RPF←RPF-1 If (A[character]::SIB)≥Y[character]; TFFF←0; RPF←RPF+1, DIB←DIB-1, (PsC)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (DC)	1	1	X1

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
VF8 (Cont)						{If (A[character]::SIB)<Y[character]; SIB←SIB+1, DIB←DIB+1} until RPZF = 1*SERR=0; (Ps)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (QC)			
PF9	CGEU	H	Compare characters greater or equal, update	0	1	Cr←Br, (Ps)←EXSU, Ar←M@SBR+SIR, TFFF←1, RPF←Cr, Br←M@DBR+DIR {Yr = Br, DIS←SIB(B≡)-DIB(B≡), TOA←DIB(B≡), DIB←DIB+1, TOM←DIB(B≡), RPF←RPF-1 If (A[character]::SIB) ≠ (B[character]::DIB); TFOF←1 If (A[character]::SIB)<(B[character]::DIB); TFFF←0, DIB←DIB+1, SIB←SIB+1} until RPZF = 1*SERR=0 If RPZF ≠ 1*SERR=0; TFOF←0, (Ps)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (QC)	1	1	X1
VF9	SGEU	H	Scan while greater or equal, update	0	1	Cr←Br, (Ps)←EXPU, Ar←M@SBR+SIR, TFFF←1, RPF←Cr, RPF←RPF-1 If (A[character]::SIB)<Y[character]; TFFF←0, RPF←RPF+1, DIB←DIB-1, (Ps)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (QC) {If (A[character]::SIB)≥Y[character]; SIB←SIB+1, DIB←DIB+1} until RPZF=1 *SERR=0; (Ps)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (QC)	1	1	X1

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
PFA	CGTU	H	Compare characters greater, update	0	1	Cr←Br, (Ps)←EXSU, Ar←M@SBR+SIR, TFFF←1, RPF←Cr, Br←M@DBR+DIR (Yr = Br, DIS←SIB(B≡)-DIB(B≡), TOA←DIB(B≡), DIB←DIB+1, TOM←DIB(B≡), RPF←RPF-1, If (A[character]::SIB ≠ (B[character]::DIB); TFOF←1 If (A[character]::SIB≤(B[character]::DIB); TFFF←0, DIB←DIB+1, SIB←SIB+1) until RPZF=1 *SERR=0; TFOF←0, (Ps)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (OC)	1	1	X1
VFA	SGTU	H	Scan while greater, update	0	1	Cr←Br, (Ps)←EXPU, Ar←M@SBR+SIR, TFFF←1, RPF←Cr, RPF←RPF-1 If (A[character]::SIB)≤(B[character]::DIB); TFFF←0, RPF←RPF+1, DIB←DIB-1, (Ps)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (OC) (If (A[character]::SIB)>(B[character]::DIB); SIB←SIB+1, DIB←DIB+1) until RPZF=1 *SERR=0; (Ps)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (OC)	1	1	X1
PFB	CLEU	H	Compare characters less or equal, update	0	1	Cr←Br, (Ps)←EXSU, Ar←M@SBR+SIR, TFFF←1, RPF←Cr, Br←M@DBR+DIR (Yr = Br, DIS←SIB(B≡)-DIB(B≡), TOA←DIB(B≡), DIB←DIB+1, TOM←DIB(B≡), RPF←RPF-1	1	1	X1

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS		OPERATOR ACTION	STACK AFTER OPERATION		HARDWARE FLOW CHART
				AROF	BROF		AROF	BROF	
PFB (Cont)						if (A[character]::SIB) ≠ (B[character]::DIB); TFOF←1 if (A[character]::SIB)>(B[character]::DIB); TFFF←0 DIB←DIB+1, SIB←SIB+1 until RPZF=1 *SERR=0; TFOF←0, (PSD)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (OC)			
VFB	SLEU	H	Scan while less or equal, update	0	1	Cr←Br, (PSD)←EXPU, Ar←M@SBR+SIR, TFFF←1, RPF←Cr, RPF←RPF-1, if (A[character]::SIB)>Y[character]; TFFF←0, RPF←RPF+1, DIB←DIB-1, (PSD)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (OC) {if (A[character]::SIB)≤Y[character]; SIB←SIB+1, DIB←DIB+1} until RPZF=1 *SERR=0; (PSD)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (OC)	1	1	X1
PFC	CEQU	H	Compare characters equal, update	0	1	Cr←Br, (PSD)←EXSU, Ar←M@SBR+SIR, TFFF←1, RPF←Cr, Br←M@DBR+DIR, {Yr = Br, DIS←SIB(B≡)-DIB(B≡), TOA←DIB(B≡), DIB←DIB+1, TOM←DIB(B≡), RPF←RPF-1 if (A[character]::SIB) ≠ (B[character]::DIB); TFOF←1 and TFFF←0, DIB←DIB+1, SIB←SIB+1} until RPZF=1 *SERR=0; TFOF←0, (PSD)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (OC)	1	1	X1

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
VFC	SEQU	H	Scan while equal, update	0	1	Cr←Br, (PsC)←EXPU, Ar←M@SBR+SIR, TFFF←1, RPF←Cr, RPF←RPF-1 If (A[character]::SIB ≠ Y[character]; TFFF←0, RPF←RPF+1, DIB←DIB-1 (PsC)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (QC) {If (A[character]::SIB) = Y[character]; SIB←SIB+1, DIB←DIB+1} until RPZF=1 *SERR=0; (PsC)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr and (QC)	1	1	X1
PFD	CNEU	H	Compare characters not equal, update	0	1	Cr←Br, (PsC)←EXSU, Ar←M@SBR+SIR, TFFF←1, RPF←Cr, Br←M@DBR+DIR {Yr = Br, DIS←SIB(B≡)-DIB(B≡), TOA←DIB(B≡), DIB←DIB+1 TOM←DIB(B≡), RPF←RPF-1 If (A[character]::SIB) ≠ (B[character]::DIB); TFOF←1 If (A[character]::SIB) = (B[character]::DIB); TFFF←0 DIB←DIB+1, SIB←SIB+1} until RPZF=1 *SERR=0; TFOF←0 (PsC)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (QC)	1	1	X1
VFD	SNEU	H	Scan while not equal update	0	1	Cr←Br, (PsC)←EXPU, Ar←M@SBR+SIR, TFFF←1, RPF←Cr, RPF←RPF-1	1	1	XL

HEX CODE	MNEMONIC	FAMILY	OPERATOR	NECESSARY STACK CONDITIONS AROF	NECESSARY STACK CONDITIONS BROF	OPERATOR ACTION	STACK AFTER OPERATION AROF	STACK AFTER OPERATION BROF	HARDWARE FLOW CHART
VFD (Cont)						If (A[character]::SIB) = Y[character]; TFFF←0, RPF←RPF+1; DIB←DIB+1, (Ps)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr, and (QC) {If (A[character]::SIB) ≠ Y[character]; SIB←SIB+1, DIB←DIB+1} until RPZF=1 *SERR=0; (Ps)←ENDE, ADJ(0,0), Br←RPF, Ar←Yr and (QC)			
UFE	NOOP	H	No operation	-	-	SECL←1, and (QC)	-	-	X7
UFF	NVLD	H	Invalid operator	-	-	QFD1←1 If EDIT = 0; invalid operator interrupt occurs If EDIT = 1; RPF←Cr, ENDE←1, and invalid operator interrupt occurs	-	-	X8

OPERATOR HEXADECIMAL CODE ASSIGNMENTS

1st \ 2nd → PRIMARY	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	FAMILY	
0	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	0	J
1	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	1	J
2	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	2	J
3	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	VALC	3	J
4	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	4	K
5	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	5	K
6	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	6	K
7	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	NAMC	7	K
8	ADD	SUBT	MULT	DIVD	IDIV	RDIV	MYIA	BTGR	LESS	GREQ	GRTR	LSEQ	EQUL	NEQL	CMSW	MULX	8	A
9	LAND	LOR	LNOT	LEQV	SAME	VARI	BSET	BRST	FLTR	DFTR	ISOL	DISO	INSR	DINS	BRST	DBRS	9	B
A	BRFL	BRTR	BRUN	EXIT	STBR	NXLR	INDX	RETN	DBFL	DBTR	DBUN	ENTR	EVAL	NXLV	HKST	STFF	A	C
B	ZERO	ONE	LT8	LT16	PUSH	DLET	EXCH	DUPL	STOD	STON	OVRD	OVRN		LOAD	LT48	MPCW	B	D
C	SCLF	DSLF	SCRT	DSRT	SCRS	DSRS	SCRF	DSRF	SCRR	DSRR	ICVD	ICVB	SNGF	SNGL	XFND	IMKS	C	E
D	TEED	PACD	EXSD	TWSD	TWOD	SISD	SXSN	RDFF	TEEU	PACU	EXSU	TWSU	TWOU	EXPU	RTFF	HALT	D	F
E	TLSD	TGED	TGTD	TLED	TEQD	TNED	TUNO		TLSU	TGEU	TGTU	TLEU	TEQU	TNEU	TUNU		E	G
F	CLSD	CGED	CGTD	CLED	CEQD	CNED			CLSU	CGEU	CGTU	CLEU	CEQU	CNEU	NOOP	NVLD	F	H

PRINTED IN U.S. AMERICA
5000276

OPERATOR HEXADECIMAL CODE ASSIGNMENTS (Cont)

VARIANT

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	FAMILY	
4			JOIN	SPLT	IDLE	SINT	REXI	DEXI			SCNI	SCNO			WHOI	NEVU	4	C
8						DCRX		NTGD				LOG2					8	A
A															INVL	PCNL	A	C
0					STAC	RTAG	RSUP	RSDN	RPRR	SPRR	RDLK	CBDN	LODT	LLLH	SRCH		0	D
D	USMD	UABD	TWFD	TWTD	SWFD	SWTD		TRNS	USNU	UABU	TWFU	TWTU	SWFU	SWTU		HALT	D	F
E																	E	G
F	SLSD	SGED	SGTD	SLED	SEQD	SNED			SLSU	SGEU	SGTU	SLEU	SEQU	SNEU	NOOP	NVLD	F	H

EDIT

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	FAMILY	
D	MINS	XFLT	SFSC	SNSC	RSTF	ENDF	MVNU	MCHR	INOP	INSG	SFDC	SRDC	INSU	INSC	ENDE	HALT	D	F
F															NOOP	NVLD	F	H

SECTION 2

MDL OPERATORS

SECTION 2 — CONTENTS

The terms "Multiplexor" and "I/O Processor" are synonymous.

SECTION 2

MDL OPERATORS

A[8:3]	=	indicates three consecutive bits with A08F being the most significant
A[8:5] = 00110	=	indicates the binary value assigned to five bits with the left digit being the most significant
"0","1", or "2"	=	binary values designates a given bit to be "false", "true", or "insignificant" respectively
@	=	addressed by
+	=	indicates an "OR" function
*	=	indicates an "AND" function
[ ]	=	contents of
←	=	set to or replaced by

MNE	OP CODE	OPERATION
SCI	A[2:2]=10	Sets 8 flip-flops @ C[8:8] to value specified in A[10:8]
STI	A[2:2]=01	Sets 8 flip-flops @ C[8:8] to value specified in A[10:8], emit clock pulse 200 nanoseconds later if OISF=0 or 600 nanoseconds later if OISF=1.
TSS	A[6:6]=001000	Emits clock pulse, compares output of one of eight flip-flops selected by A[9:3] @ C[8:8] to the state of A[10:1]. Sets ERR1 or ERR2 and halts processor operation if miscomparison occurs.
TSR	A[6:6]=101000	Emits clock pulse, compares output of one of eight flip-flops selected by A[9:3] @ C[8:8] to the state of A[10:1]. Resets ERR1 or ERR2 if a comparison occurs.
SSE	A[6:6]=011000	Compares output of one of eight flip-flops selected by A[9:3] @ C[8:8] to the state of A[10:1]. Sets ERR1 or ERR2 if a miscomparison occurs.

MDL OPERATORS

MNE	OP CODE	OPERATION
SRE	A[6:6]=111000	Compares output of one of eight flip-flops selected by A[9:3] @ C[8:8] to the state of A[10:1]. Resets ERR1 or ERR2 if a miscomparison occurs.
END	A[3:3]=100	If (ERR1 + ERR2 + HLTF) = 1 and CYCM=0, ERR1←0, ERR2←0, and MDL operation is halted displaying string number in A[10:7] and C[8:8]. If (ERR1 * ERR2 * HLTF) = 0, then a NO-OP occurs.
BRZ1	A[8:8]=110X0000	Inhibits output of MA to cause the next command to be fetched from address 0.
CPM	A[8:8]=001X0000	Forces ENR command if ERR1=1, otherwise becomes a NO-OP.
ENR	A[8:8]=01000000	Clears MDL registers and changes MDL to I/O operation
SAD	A[8:8]=10000000	If CYCM=1 then MA[10:2]←A[10:2] and MA[8:8]←C[8:8] If CYCM=0 then a NO-OP occurs.

SECTION 3

DATA COMMUNICATIONS PROCESSOR INSTRUCTIONS

SECTION 3 — CONTENTS

The terms "Multiplexor" and "I/O Processor" are synonymous.

SECTION 3

DATA COMMUNICATIONS PROCESSOR INSTRUCTIONS

DCP INSTRUCTION LIST (BY GROUP)

Group	Op Code	A Field	MNE	Operation
0	00000	000	IDLE	Idle
	00010	0VV	AWI	Adapter Write
	00010	1VV	AWRR	Adapter Interrogate
	00001	VVV	BKP	Stop/Branch on Breakpoint
	00011	100	SHFT	Shift MA Registers Right
	00011	101	HEYU	Set System Interrupt
	00011	000	MSKW	Set Cluster Mask Register
	00011	001	MSKR	Read Cluster Mask Register
1	00100	000	ARWN	Adapter Read When Needed
	00100	001	ARIN	Adapter Read If Needed
	00100	010	DBYZ	Decrement Branch Y Is Zero
	00100	011	DBYN	Decrement Branch Y Is Non-Zero
	00100	100	BRAN	Branch If System Attention Needed
	00101	VVV	GOTO	Branch
	00110	VVV	GOX	Branch Relative with X
	00111	VVV	GOI	Branch Relative Indirect
2	01000	000	MOVE	Move Halfword
	01000	001	LMRI	Local Memory Read Indirect
	01000	010	SMRD	Scratchpad Memory Read Direct
	01000	011	LMRD	Local Memory Read Direct
	01000	101	LMWI	Local Memory Write Indirect

DCP INSTRUCTIONS

DCP INSTRUCTION LIST (BY GROUP) (Cont)

Group	Op Code	A Field	MNE	Operation
2	01000	110	SMWD	Scratchpad Memory Write Direct
	01000	111	LMWD	Local Memory Write Direct
	01001	001	MMR	Main Memory Read
	01001	100	MMWU	Main Memory Write Unconditional
	01001	101	MMWR	Main Memory Write Unconditional Retain Readout
	01001	110	MMWP	Main Memory Protected Write
	01001	111	MMPR	Main Memory Protected Write Retain Readout
3	01100	000	HAD	Halfword Add
	01101	000	HADB	Halfword Add B Literal
	01100	100	HSB	Halfword Subtract
	01101	100	HSBB	Halfword Subtract B Literal
	01110	VVV	TRAN	Translate
	01111	00V	PARY	Parity
4	11101	AAA	ADD	Add
	11110	AAA	ADDB	Add B Literal
	11111	AAA	ADDC	Add C Literal
	10000	AAA	SUB	Subtract
	11000	AAA	SUBB	Subtract B Literal
	11100	AAA	SUBC	Subtract C Literal
	10101	AAA	LAN	Logical AND
	10110	AAA	LANB	Logical AND B Literal
	10111	AAA	LANC	Logical AND C Literal
	10100	AAA	LAOM	Logical AND-OR MA

DCP INSTRUCTION LIST (BY GROUP) (Cont)

Group	Op Code	A Field	MNE	Operation
4	10001	AAA	LOR	Logical OR
	10010	AAA	LORB	Logical OR B Literal
	10011	AAA	LORC	Logical OR C Literal
	11001	AAA	LEO	Logical Exclusive OR
	11010	AAA	LEOB	Logical Exclusive OR B Literal
	11011	AAA	LEOC	Logical Exclusive OR C Literal

DCP INSTRUCTION LIST (BY MNEMONIC)

MNE	Operation	Group	Op Code	A Field
ADD	Add	4	11101	AAA
ADDB	Add B Literal		11110	AAA
ADDC	Add C Literal		11111	AAA
ARIN	Adapter Read If Needed	1	00100	001
ARWN	Adapter Read When Needed	1	00100	000
AWI	Adapter Write	0	00010	0VV
AWRR	Adapter Interrogate	0	00010	1VV
BKP	Stop/Branch on Breakpoint	0	00001	VVV
BRAN	Branch If System Attention Needed	1	00100	100
DBYN	Decrement Branch Y Is Non-Zero	1	00100	011
DBYZ	Decrement Branch Y Is Zero	1	00100	010
GOI	Branch Relative Indirect	1	00111	VVV
GOTO	Branch	1	00101	VVV

DCP INSTRUCTIONS

DCP INSTRUCTION LIST (BY MNEMONIC) (Cont)

MNE	Operation	Group	Op Code	A Field
GOX	Branch Relative with X	1	00110	VVV
HAD	Halfword Add	3	01100	000
HADB	Halfword Add B Literal	3	01101	000
HEYU	Set System Interrupt	0	00011	101
HSB	Halfword Subtract	3	01100	100
HSBB	Halfword Subtract B Literal	3	01101	100
IDLE	Idle	0	0000	000
LAN	Logical AND	4	10101	AAA
LANB	Logical AND B Literal	4	10110	AAA
LANC	Logical AND C Literal	4	10111	AAA
LAOM	Logical AND-OR MA	4	10100	AAA
LEO	Logical Exclusive OR	4	11001	AAA
LEOB	Logical Exclusive OR B Literal	4	11010	AAA
LEOC	Logical Exclusive OR C Literal	4	11011	AAA
LMRD	Local Memory Read Direct	2	01000	011
LMRI	Local Memory Read Indirect	2	01000	001
LMWD	Local Memory Write Direct	2	01000	111
LMWI	Local Memory Write Indirect	2	01000	101
LOR	Logical OR	4	10001	AAA
LORB	Logical OR B Literal	4	10010	AAA
LORC	Logical OR C Literal	4	10011	AAA
MMPR	Main Memory Protected Write Retain Readout	2	01001	111
MMWP	Main Memory Protected Write	2	01001	110
MMR	Main Memory Read	2	01001	001

DCP INSTRUCTION LIST (BY MNEMONIC) (Cont)

MNE	Operation	Group	Op Code	A Field
MMWR	Main Memory Write Unconditional Retain Readout	2	01001	101
MMWU	Main Memory Write Unconditional	2	01001	100
MOVE	Move Halfword	2	01000	000
PARY	Parity	3	01111	00V
MSKR	Read Cluster Mask Register	0	00011	001
MSKW	Set Cluster Mask Register	0	00011	000
SHFT	Shift MA Registers Right	0	00011	100
SMRD	Scratchpad Memory Read Direct	2	01000	010
SMWD	Scratchpad Memory Write Direct	2	01000	110
SUB	Subtract	4	10000	AAA
SUBB	Subtract B Literal	4	11000	AAA
SUBC	Subtract C Literal	4	11100	AAA
TRAN	Translate	3	01110	VVV

DCP INSTRUCTION LIST (BY OPERATION)

Operation	MNE	Group	Op Code	A Field
Adapter Interrogate	AWRR	0	00010	1VV
Adapter Read If Needed	ARIN	1	00100	001
Adapter Read When Needed	ARWN	1	00100	000
Adapter Write	AWI	0	00010	0VV
Add	ADD	4	11101	AAA
Add B Literal	ADDB	4	11110	AAA
Add C Literal	ADDC	4	11111	AAA

DCP INSTRUCTIONS

DCP INSTRUCTION LIST (BY OPERATION) (Cont)

Operation	MNE	Group	Op Code	A Field
Branch	GOTO	1	00101	VVV
Branch If System Attention Needed	BRAN	1	00100	100
Branch Relative Indirect	GOI	1	00111	VVV
Branch Relative With X	GOX	1	00110	VVV
Decrement Branch Y Is Non-Zero	DBYN	1	00100	011
Decrement Branch Y Is Zero	DBYZ	1	00100	010
Halfword Add	HAD	3	01100	000
Halfword Add B Literal	HADB	3	01101	000
Halfword Subtract	HSB	3	01100	000
Halfword Subtract B Literal	HSBB	3	01101	100
Idle	IDLE	0	00000	000
Local Memory Read Direct	LMRD	2	01000	011
Local Memory Read Indirect	LMRI	2	01000	001
Local Memory Write Direct	LMWD	2	01000	111
Local Memory Write Indirect	LMWI	2	01000	101
Logical AND	LAN	4	10101	AAA
Logical AND B Literal	LANB	4	10110	AAA
Logical AND C Literal	LANC	4	10111	AAA
Logical AND-OR MA	LAOM	4	10100	AAA
Logical Exclusive OR	LEO	4	11001	AAA
Logical Exclusive OR B Literal	LEOB	4	11010	AAA
Logical Exclusive OR C Literal	LEOC	4	11011	AAA
Logical OR	LOR	4	10001	AAA

DCP INSTRUCTION LIST (BY OPERATION) (Cont)

Operation	MNE	Group	Op Code	A Field
Logical OR B Literal	LORB	4	10010	AAA
Logical OR C Literal	LORC	4	10011	AAA
Main Memory Protected Write	MMWP	2	01001	110
Main Memory Protected Write Retain Readout	MMPR	2	01001	111
Main Memory Read	MMR	2	01001	001
Main Memory Write Unconditional	MMWU	2	01001	100
Main Memory Write Unconditional Retain Readout	MMWR	2	01001	101
Move Halfword	MOVE	2	01000	000
Parity	PARY	3	01111	00V
Read Cluster Mask Register	MSKP	0	00011	001
Scratchpad Memory Read Direct	SMRD	2	01000	010
Scratchpad Memory Write Direct	SMWD	2	01000	110
Set Cluster Mask Register	MSKW	0	00011	000
Set System Interrupt	HEYU	0	00011	101
Shift MA Registers Right	SHFT	0	00011	100
Stop/Branch on Breakpoint	BKP	0	00001	VVV
Subtract	SUB	4	10000	AAA
Subtract B Literal	SUBB	4	11000	AAA
Subtract C Literal	SUBC	4	11100	AAA
Translate	TRAN	3	01110	VVV

UNDEFINED OPERATION CODES

Operators containing undefined operation codes are not released; they act like the Idle instruction. The following codes are exceptions to the above rule:

Operation Code	Interpreted As	Instruction
01000 100	01000 000	Move half-word
01001 000	01000 000	Move half-word
01001 01n	01000 000	Move half-word
01100 001	01100 000	Half-word Add
01100 01n	01100 000	Half-word Add
01100 101	01100 100	Half-word Subtract
01100 11n	01100 100	Half-word Subtract
01101 011	01101 000	Half-word Add, literal in the B field
01101 01n	01101 000	Half-word Add, literal in the B field
01101 101	01101 100	Half-word Subtract, literal in the B field
01101 11n	01101 100	Half-word Subtract, literal in the B field
01111 01n	11001 nnn	Exclusive OR
01111 1nn	11001 nnn	Exclusive OR

OPERATOR (INSTRUCTION SET) DESCRIPTIONS

(A)	The parentheses are used to indicate that "A" is pointing to an address located within memory.
A	Indicates that a literal value is contained within "A".
+	Indicates an arithmetic plus sign.
=	Indicates an arithmetic equal sign.
≠	Indicates not equal arithmetically.
A←B	Indicates that "A" is replaced by "B".
* or	An asterisk or a period indicates a logical AND.
: or ; or ,	A colon, semicolon, or comma indicates a separator.
v	Indicates versus.
rA	Indicates row A.
rB	Indicates row B.
Example:	
B[2:3]	Indicates three bits of the B register, starting with bit 2 and counting toward bit 0 (e.g., bits 2, 1, and 0 of the B register).

GROUP 0

Mnemonic	OP Code	"A" Field	Description
IDLE	00	0	Idle until fault interrupt (e.g., initialize or time-out)
AWI	02	0...3	A = 0: normal; CLIN←AA.AC.AI A = 1: AI←C; CLIN←AA.AC.AI A = 2: AC←B; CLIN←AA.AC.AI A = 3: AC←B; AI←C; CLIN←AA.AC.AI
AWRR	02	4...7	A = 0: normal; CLIN←AA.AC.AI; AC.AI←CLIN A = 1: AI←C; CLIN←AA.AC.AI; AC.AI←CLIN A = 2: AC←B; CLIN←AA.AC.AI; AC.AI←CLIN A = 3: AC←B; AI←C; CLIN←AA.AC.AI; AC.AI←CLIN

OPERATOR DESCRIPTIONS

GROUP 0 (Cont)

Mnemonic	OP Code	"A" Field	Description
BKP	01		B:C = branch address; MA←IA; SPO-R←MA
SHFT	03	4	B[2:3] = number of shifts -1
MEYU	03	5	B[2:3] = 4; C not used; Send interrupt signal to multiplexor
MSKW	03	0	CMR←AC.AI
MSKR	03	1	AC.AI←CMR; AA←0; B:C not used

GROUP 1

Mnemonic	OP Code	"A" Field	Description
ARWN	04	0	CAN = 0: wait CAN = 1: AA.AC.AI←CLIN; IA←B.C; CFO:1←0
ARIN	04	1	CAN = 0: end CAN = 1: AA.AC.AI←CLIN; IA←B.C; CFO:1←0
DBYZ	04	2	Y = 0: IA←B:C; CFO.CF1←0 Y ≠ 0: Y←Y -1
DBYN	04	3	Y = 0: end Y ≠ 0: IA←B:C; Y←Y -1; CFO.CF1←0
BRAN	04	4	MSAN = 0: end MSAN = 1: IA←B:C; CFO:1←0
GOTO	05		No match: end If match: IA←B.C CFO.CF1←0
GOX	06		No match: end If match: IA←B.C, CFO:CF1←0; IAO←IAO+X, IAI←IAI+1
GOI			A = used for the branch condition If match: IA←B.C; B←Y; CFO:1←0; IAO←IAO+(8)

GROUP 2

Mnemonic	OP Code	"A" Field	Description
MOVE	08	0	(C)←MA; MA←(B)
LMRD	08	3	MA←B.C; W←LM(MA)
LMRI	08	1	(C)←MA; MA←(B) W←LM(MA)
LMWD	08	7	MA←B.C; LM(MA)←W
LMWI	08	5	(C)←MA; MA←(B) LM(MA)←W
SMRD	08	2	(C)←MA; W←SM(B) or R(B)
SMWD	08	6	SM(C)←W or R(C)←W -1; MA←(B)
MMR	09	1	(C)←MA; MA←(B); W←MM(MA)
MMWU	09	4	(C)←MA; MA←(B); MM(MA)←W
MMWR	09	5	(C)←MA; MA←(B); MM(MA)←W; W←MM(MA)
MMWP	09	6	(C)←MA; MA←(B); MM(MA)←W only if bit 48 = 0
MWRP	09	7	(C)←MA; MA←(B); W←MM(MA); MM(MA)←W only if bit 48 = 0

GROUP 3

Mnemonic	OP Code	"A" Field	Description
HAD	0C	0	(C) all rows←MA all rows + (B) all rows; CF0←carry; CF1←all sums zero
HADB	0D	0	(C) all rows←MA all rows +B; CF0←carry; CF1←all sums zero
HSB	0C	4	(C) all rows←MA all rows - (B) all rows; CF0←borrow; CF1← all remainders zero

OPERATOR DESCRIPTIONS

GROUP 3 (Cont)

Mnemonic	OP Code	"A" Field	Description
HSBB	0D	5	(C) all rows←MA all rows -B;
TRAN	0E		(C)←translator←(B); A = type of translation
PARY	0F	0+	(C)←(B) and fix parity A = 0: odd A = 1: even

GROUP 4

Mnemonic	OP Code	"A" Field	Description
ADD	1D		(C)←(A)+(B)+CF0
ADDB	1E		(C)←(A)+B+CF0
ADDC	1F	A ≠ 1	(A)←(B)+C+CF0; (B)←(B)+C+CF0;
SUB	10		(C)←(A) -(B) -CF0; CF0←borrow out; CF1←remainder zero;
SUBB	18		(C)←(A) -B -CF0; CF0←borrow out; CF1←remainder zero
SUBC	1C		A ≠ 1: (A)←C - (B) - CF0; CF0←borrow out; CF1←remainder zero A = 1: (B)←C - (B) - CF0; CF0←borrow out; CF1←remainder zero
LAN	15		(C)←(A) * (B); CF0←0; CF1←result zero
LANB	16		(C)←(A) * B; CF0←0; CF1←result zero
LANC	17		A ≠ 1: (A)←C * (B); CF0←0; CF1←result zero A = 1: (B)←C * (B); CF0←0; CF1←result zero
LAOM	14		A ≠ 1: (A)←[C * (B)] or MA(rA) A = 1: (B)←[C * (B)] or MA(rB) CF0←0; CF1←result zero;

GROUP 4 (Cont)

Mnemonic	OP Code	"A" Field	Description
LOR	11		(C)←(A) or (B) CF0←0; CF1←result zero
LORB	12		(C)←(A) or (B) CF0←0; CF1←result zero
LORC	13		A ≠ 1: (A)←(B) or C; CF0←0 A = 1: (B)←(B) or C; CF1←result zero
LEO	19		(C)←(A) + (B); CF0←0; CF1←result zero
LEOB	1A		(C)←(A) + (B); CR0←0; CF1←result zero
LEOC	1B		A ≠ 1: (A)←C + (B); CF0←0; CF1←result zero A = 1: (B)←C + (B); CF0←0; CF1←result zero

SECTION 4

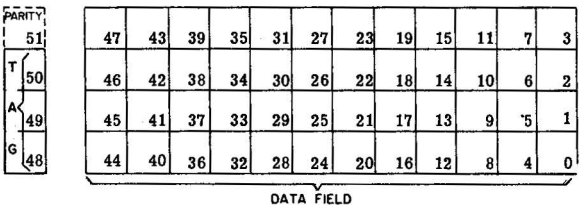
WORD FORMATS

The terms "Multiplexor" and "I/O Processor" are synonymous.

SECTION 4

WORD FORMATS

BASIC WORD STRUCTURE



51 - Parity bit (odd parity)

50 => 48 - Tag field

47 => 0 - Data field

The processors and multiplexors of the B 6700 primarily handle "words" of information. Each word is made up of a data field, a tag field, and a parity bit. The tag field is the major factor determining the meaning of the data field.

DESCRIPTORS

PRESENCE AND COPY BIT MEANINGS

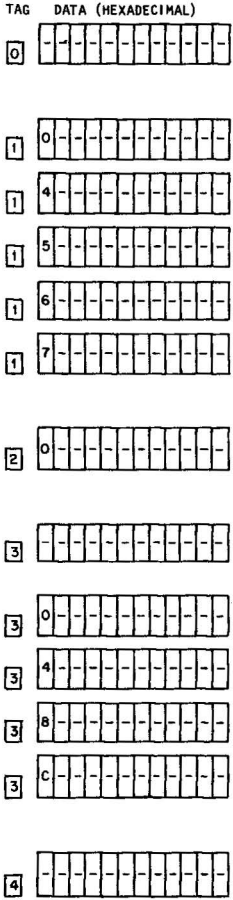
MOM (ORIGINAL) DESCRIPTOR

47 Presence	46 Copy	Meaning
0	0	Data/code-segment absent; Disk address in [19:20], Size in [39:20].
1	0	Data/code-segment present; Core address in [19:20], Size in [39:20].

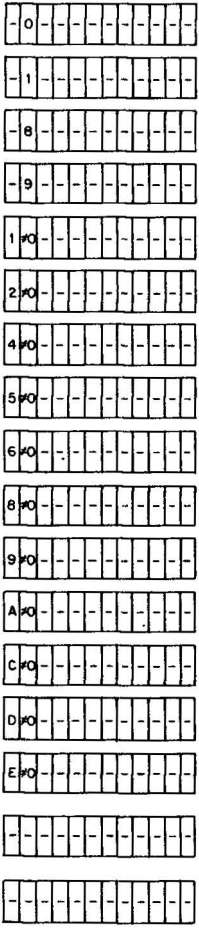
COPY DESCRIPTOR

47 Presence	46 Copy	Meaning
0	1	Data/code-segment may or may not be present. See presence bit in original. Core address of MOM descriptor in [19:20].
1	1	Data/code segment present. Core address in [19:20].

WORD IDENTIFICATION

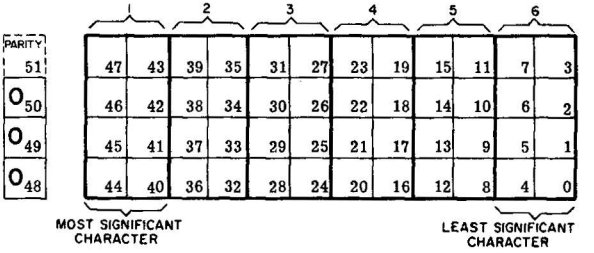


WORD IDENTIFICATION

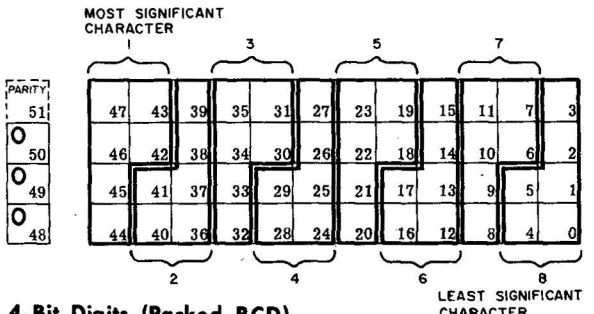


CHARACTER REPRESENTATION

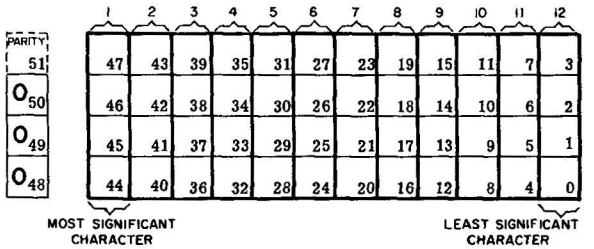
8-Bit Bytes (EBCDIC Code)



6-Bit Characters (BCL Code)



4-Bit Digits (Packed BCD)



SIGNS OF NUMERIC FIELDS

The sign of a numeric field is represented as follows:

a. 8-bit characters

   The sign is in the zone bits of the least significant character (bits 7 thru 4 of the field). A bit configuration of 1101 indicates a negative number; any other bit configuration indicates a positive number.

b. 6-bit characters

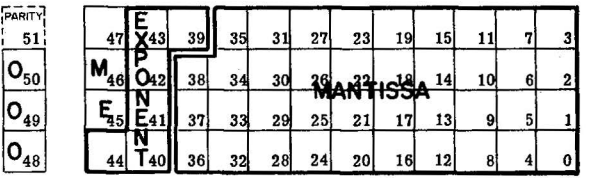
   The sign is in the zone bits of the least significant character (bits 5 and 4 of the field). A bit configuration of 10 indicates a negative number; any other bit configuration indicates a positive number.

c. 4-bit digits

   The sign is carried as a separate digit, and it is the most significant digit of the field. A bit configuration of 1101 indicates a negative number; any other bit configuration indicates a positive number.

TAG = 0

SINGLE PRECISION OPERAND



Tag - 0

47 - Not used

M - Sign of mantissa - 1 = negative, 0 = positive

E - Sign of exponent - 1 = negative, 0 = positive

44 => 39 - Exponent

38 => 0 - Mantissa

TAG = 0

OCCURS INDEX WORD

(A Register)

PARITY 51	47	43	39	35	31	27	23	19	15	11	7	3
O_{50}	46	42	38	34	30	26	22	18	14	10	6	2
O_{49}	LENGTH 45	41	37	33	29	SIZE 25	21	17	13	OFFSET 9	5	1
O_{48}	44	40	36	32	28	24	20	16	12	8	4	0

47 => 32 = Length field

31 => 16 = Size field

15 => 0 = Offset field

Result of OCRX Operator

(B Register)

The OIW and the operand in the B register are used to calculate a new index value. This new value is placed in the B register.

TAG = 0

SCAN IN AND SCAN OUT FUNCTION WORDS

PARITY 51	47	43	39	35	31	27	23	O_{19}	U_{15}	11	F_7	Z_3
O_{50}	46	42	38	34	30	26	22	O_{18}	N_{14}	10	F_6	Z_2
O_{49}	45	41	37	33	29	25	21	O_{17}	I_{13}	9	F_5	Z_1
O_{48}	44	40	36	32	28	24	20	16	T_{12}	F_8	O_4	M_0

16 => 9 = Unit Designate Field - Unique for each unit and installation.

F BITS (SCAN IN)

8765	OPERATION
0000	Interrogate I/O Path for Upcoming Initiate I/O Operation
0001	Interrogate Peripheral Status of the Designated Status Vector
0010	Read Result Descriptor
0011	Read Time of Day Register
0100	Read Interrupt Register or Read Interrupt Mask Register, or Read General Control Adapter
0110	Interrogate Peripheral Unit Type
1111	Read Interrupt Literal

F BITS (SCAN OUT)

8765	OPERATION
0000	Initiate I/O Operation
0011	Set Time of Day Register
0100	Set Interrupt Mask Register
0101	Set General Control Adapter

Z = Multiplexor Field: 001 = MPX A, 010 = MPX B, 100 = MPX C (if M = 1)

M = 0, all multiplexors respond; M = 1, MPX designated by Z responds

TAG = 0

INTERROGATE I/O PATH FUNCTION WORD

(A Register)

PARITY 51	47	43	39	35	31	27	23	O_{19}	U_{15}	11	O_7	Z_3
O_{50}	46	42	38	34	30	26	22	O_{18}	N_{14}	10	O_6	Z_2
O_{49}	45	41	37	33	29	25	21	O_{17}	I_{13}	9	O_5	Z_1
O_{48}	44	40	36	32	28	24	20	16	T_{12}	O_8	O_4	M_0

UNIT = Unique within each installation.

Z = 001, MPX A (M = 1)

= 010, MPX B (M = 1)

= 100, MPX C (M = 1)

M = 0, All multiplexors respond

M = 1, Multiplexor designated by Z to respond

I/O PATH RESULT WORD RETURNED

(B Register)

PARITY 51	47	43	39	35	31	27	23	O_{19}	U_{15}	11	O_7	Z_3
O_{50}	46	42	38	34	30	26	22	O_{18}	N_{14}	10	O_6	Z_2
O_{49}	45	41	37	33	29	25	21	O_{17}	I_{13}	9	O_5	Z_1
O_{48}	44	40	36	32	28	24	20	16	T_{12}	O_8	O_4	A_0

UNIT = Unique within each installation
Z = xx1, Available path via MPX A
= x1x, Available path via MPX B
= 1xx, Available path via MPX C
(x = don't care)
A = 0, No path available
A = 1, Path is available

TAG = 0

INTERROGATE PERIPHERAL STATUS FUNCTION WORD

(A Register)

PARITY 51	47	43	39	35	31	27	23	O_{19}	O_{15}	N_{11}	O_7	Z_3
O_{50}	46	42	38	34	30	26	22	O_{18}	O_{14}	N_{10}	O_6	Z_2
O_{49}	45	41	37	33	29	25	21	O_{17}	O_{13}	N_9	I_5	Z_1
O_{48}	44	40	36	32	28	24	20	O_{16}	N_{12}	O_8	O_4	M_0

N = 0 through 7, Status Vector Word Number

N = 8 Status Change Vector

Z = 001, MPX A (M = 1)
= 010, MPX B (M = 1)
= 100, MPX C (M = 1)

M = 0, All multiplexors are to respond
1, Multiplexor designated by Z is to respond

STATUS VECTOR WORD RETURNED

(B Register)

PARITY 51	47	43	39	35	31	27	23	19	15	11	7	3
O_{50}	46	42	38	34	30	26	22	STATUS 18	14	10	6	2
O_{49}	45	41	37	33	29	25	21	BITS 17	13	9	5	1
O_{48}	44	40	36	32	28	24	20	16	12	8	4	X_0

32 => 1 = Status Bits

Vector word 0 displays status of peripheral units 0 thru 31
Vector word 1 displays status of peripheral units 32 thru 63
Vector word 2 displays status of peripheral units 64 thru 95
Vector word 3 displays status of peripheral units 96 thru 127
Vector word 4 displays status of peripheral units 128 thru 159
Vector word 5 displays status of peripheral units 160 thru 191
Vector word 6 displays status of peripheral units 192 thru 223
Vector word 7 displays status of peripheral units 224 thru 255
Vector word 8 displays status of status change vector

X = 0, Status Word Not Present
1, Status Word Present

TAG = 0

READ RESULT DESCRIPTOR FUNCTION WORD

(A Register)

PARITY 51	47	43	39	35	31	27	23	O_{19}	O_{15}	O_{11}	O_7	Z_3
O_{50}	46	42	38	34	30	26	22	O_{18}	O_{14}	O_{10}	I_6	Z_2
O_{49}	45	41	37	33	29	25	21	O_{17}	O_{13}	O_9	O_5	Z_1
O_{48}	44	40	36	32	28	24	20	O_{16}	O_{12}	O_8	O_4	I_0

Z = 001, MPX A

= 010, MPX B

= 100, MPX C

RESULT DESCRIPTOR RETURNED

(B Register)

PARITY 51	47	43	39	35	31	CC 27	UNIT 23	NUMBER 19	15	11	7	3
O_{50}	46	MEMORY 42	38	34	30	CC 26	INT 22	MB 18	14	ERROR 10	6	2
O_{49}	45	ADDRESS 41	37	33	29	CC 25	21	ER 17	13	FIELD 9	5	1
O_{48}	44	40	36	32	28	24	20	16	12	8	4	0

47 => 28 = Memory address of last word accessed.

27 => 25 = Character index within last word

24 => 17 = Unit number

16 => 0 = Error field

Bit 0 = Exception — Bit 4 = Descriptor Error

Bit 1 = Attention — Bit 5 = Memory Address Error

Bit 2 = Busy — Bit 6 = Memory Parity Error

Bit 3 = Not ready — Bit 16 = Memory Protect

(See individual result descriptor formats in Section 5)

TAG = 0

READ TIME-OF-DAY FUNCTION WORD

(A Register)

PARITY 51	47	43	39	35	31	27	23	O_{19}	O_{15}	O_{11}	O_7	O_3
O_{50}	46	42	38	34	30	26	22	O_{18}	O_{14}	O_{10}	I_6	O_2
O_{49}	45	41	37	33	29	25	21	O_{17}	O_{13}	O_9	I_5	O_1
O_{48}	44	40	36	32	28	24	20	O_{16}	O_{12}	O_8	O_4	O_0

TIME OF DAY WORD RETURNED

(B Register)

PARITY 51	47	43	39	35	31	27	23	19	15	11	7	3
O_{50}	46	42	38	34	30	26	22	18	14	10	6	2
O_{49}	45	41	37	33	29	25	21	17	13	9	5	1
O_{48}	44	40	36	32	28	24	20	16	12	8	4	0

TIME OF DAY (BINARY)

(Expressed in increments of 2.4 microseconds.)

TAG = 0

READ INTERRUPT REGISTER FUNCTION WORD

(A Register)

PARITY 51	47	43	39	35	31	27	23	O_{19}	O_{15}	O_{11}	I_7	Z_3
O_{50}	46	42	38	34	30	26	22	O_{18}	O_{14}	O_{10}	O_6	Z_2
O_{49}	45	41	37	33	29	25	21	O_{17}	O_{13}	O_9	O_5	Z_1
O_{48}	44	40	36	32	28	24	20	O_{16}	O_{12}	O_8	O_4	I_0

Z = 001, MPX A

= 010, MPX B

= 100, MPX C

INTERRUPT REGISTER WORD RETURNED

(B Register)

PARITY 51	47	43	39	35	31	27	23	19	15	11	7	3
O_{50}	46	42	38	34	30	26	22	18	14	10	6	2
O_{49}	45	41	37	33	29	25	21	17	13	INTERRUPT 9	5	1
O_{48}	44	40	36	32	28	24	20	16	12	REGISTER 8	4	0

9 => 0 = Interrupt Register Field

Bit 0 = Status Change
Bit 1 = DCP 1
Bit 2 = DCP 2
Bit 3 = DCP 3
Bit 4 = DCP 4
Bit 5 = External MPX
Bit 9 = I/O Finish

} 1 = interrupt pending

TAG = 0

READ INTERRUPT MASK FUNCTION WORD

(A Register)

PARITY 51	47	43	39	35	31	27	23	O_{19}	O_{15}	O_{11}	I_7	Z_3
O_{50}	46	42	38	34	30	26	22	O_{18}	O_{14}	O_{10}	O_6	Z_2
O_{49}	45	41	37	33	29	25	21	O_{17}	O_{13}	I_9	O_5	Z_1
O_{48}	44	40	36	32	28	24	20	O_{16}	O_{12}	O_8	O_4	I_0

Z = 001, MPX A

= 010, MPX B

= 100, MPX C

INTERRUPT MASK WORD RETURNED

(B Register)

PARITY 51	47	43	39	35	31	27	23	19	15	11	7	3
O_{50}	46	42	38	34	30	26	22	18	14	10	6	2
O_{49}	45	41	37	33	29	25	21	17	13	MASK 9	5	1
O_{48}	44	40	36	32	28	24	20	16	12	8	4	0

9*=> 0 = Mask field

Bit 0 = Status Change
Bit 1 = DCP 1
Bit 2 = DCP 2
Bit 3 = DCP 3
Bit 4 = DCP 4
Bit 5 = External MPX
Bit 9 = I/O Finished

} 1 = interrupt inhibited

TAG = 0

READ GENERAL CONTROL ADAPTER FUNCTION WORD

(A Register)

PARITY 51	47	43	39	35	31	27	23	O_{19}	O_{15}	O_{11}	I_7	Z_3
O_{50}	46	42	38	34	30	26	22	O_{18}	O_{14}	N_{10}	O_6	Z_2
O_{49}	45	41	37	33	29	25	21	O_{17}	O_{13}	N_9	I_5	Z_1
O_{48}	44	40	36	32	28	24	20	O_{16}	O_{12}	O_8	Z_4	I_0

Z = 0001, GCA A

Z = 0010, GCA B

Z = 0100, GCA C

Z = 1000, GCA D

N = 00, Input Register

N = 01, Interrupt Mask

N = 10, Interrupt Register

N = 11, Output Register

RESPONSE TO READ GCA

The contents of the specified General Control Adapter register are placed in the B register.

TAG = 0

INTERROGATE PERIPHERAL UNIT TYPE FUNCTION WORD (A Register)

PARITY 51	47	43	39	35	31	27	23	O_{19}	U_{15}	11	I_7	Z_3
O_{50}	46	42	38	34	30	26	22	O_{18}	N_{14}	10	I_6	Z_2
O_{49}	45	41	37	33	29	25	21	O_{17}	I_{13}	9	O_5	Z_1
O_{48}	44	40	36	32	28	24	20	16	T_{12}	O_8	O_4	M_0

16 => 9 = Unit, unique within each installation
Z = 001, MPX A (M=1)
= 010, MPX B (M=1)
= 100, MPX C (M=1)
M = 0, All Multiplexors to Respond
1, Multiplexor Designated By Z to Respond

UNIT TYPE WORD RETURNED (B Register)

PARITY 51	47	43	39	35	31	27	23	19	15	11	7	T_3
O_{50}	46	42	38	34	30	26	22	18	14	10	6	Y_2
O_{49}	45	41	37	33	29	25	21	17	13	9	5	P_1
O_{48}	44	40	36	32	28	24	20	16	12	8	4	E_0

Bits 5 => 0 = Unit type

	HEX	DECIMAL	
00 0000	(00)	0	No unit
00 0001	(01)	1	Disk file (IA and IC)
00 0010	(02)	2	Display
00 0011	(03)	3	Disk file (IIB)
00 0100	(04)	4	Paper tape reader
00 0101	(05)	5	Paper tape punch
00 0110	(06)	6	Buffered line printer, BCL drum
00 0111	(07)	7	Unbuffered line printer, BCL drum
00 1001	(09)	9	Card Reader
00 1011	(0B)	11	Card punch
00 1101	(0D)	13	Magnetic tape, 7 track, with status vector information
00 1110	(0E)	14	Magnetic tape, 9 track, NRZ, with status vector information
00 1111	(0F)	15	Magnetic tape, 9 track, PE, with status vector information
01 1101	(1D)	29	Magnetic tape, 7 track, no status vector information
01 1110	(1E)	30	Magnetic tape, 9 track, NRZ, no status vector information
01 1111	(1F)	31	Magnetic tape, 9 track, PE, no status vector information
10 0001	(21)	33	Disk file (IA and IC) connected to DF0
10 0011	(23)	35	Disk file (IIB) connected to DF0
10 0110	(26)	38	Buffered line printer, EBCDIC subset drum
10 0111	(27)	39	Unbuffered line printer, EBCDIC subset drum

TAG = 0

READ INTERRUPT LITERAL FUNCTION WORD

(A Register)

PARITY 51	47	43	39	35	31	27	23	O_{19}	O_{15}	O_{11}	I_7	Z_3
O_{50}	46	42	38	34	30	26	22	O_{18}	O_{14}	O_{10}	I_6	Z_2
O_{49}	45	41	37	33	29	25	21	O_{17}	O_{13}	O_9	I_5	Z_1
O_{48}	44	40	36	32	28	24	20	O_{16}	O_{12}	I_8	O_4	I_0

Z = 001, MPX A

= 010, MPX B

= 100, MPX C

INTERRUPT LITERAL WORD RETURNED

(B Register)

PARITY 51	47	43	39	35	31	27	23	19	15	11	INTERRUPT 7	LITERAL 3
O_{50}	46	42	38	34	30	26	22	18	14	10	6	2
O_{49}	45	41	37	33	29	25	21	17	13	9	5	1
O_{48}	44	40	36	32	28	24	20	16	12	8	4	0

Bits 7 => 4, 0001 = DCP 1
0010 = DCP 2
0011 = DCP 3
0100 = DCP 4
1001 = Multiplexor I/O finished
1111 = Status change

Bits 3 => 0, 0001 = MPX A
0010 = MPX B
0100 = MPX C

TAG = 0

INITIATE I/O FUNCTION WORD

(A Register)

PARITY												
51	47	43	39	35	31	27	23	O_{19}	U_{15}	11	O_7	Z_3
O_{50}	46	42	38	34	30	26	22	O_{18}	N_{14}	10	O_6	Z_2
O_{49}	45	41	37	33	29	25	21	O_{17}	I_{13}	9	O_5	Z_1
O_{48}	44	40	36	32	28	24	20	16	T_{12}	O_8	O_4	I_0

Z = 001, MPX A

= 010, MPX B

= 100, MPX C

16 => 9 = Unit Designate (unique for each installation)

AREA DESCRIPTOR SENT TO MULTIPLEXOR

(B Register)

PARITY												
51	47	43	39	35	31	27	23	19	15	11	7	3
O_{50}	46	42	CHAR 38	34	WORD 30	26	22	18	AREA BASE 14	10	6	2
O_{49}	45	41	CT 37	33	COUNT 29	25	21	17	ADDRESS 13	9	5	1
O_{48}	44	40	36	32	28	24	20	16	(IOCW) 12	8	4	0

39 = 37 = Character count

36 = 20 = Word Count

19 = 0 = Area base address

TAG = 0

SET TIME OF DAY CLOCK FUNCTION WORD

(A Register)

PARITY 51	47	43	39	35	31	27	23	O_{19}	O_{15}	O_{11}	O_7	O_3
O_{50}	46	42	38	34	30	26	22	O_{18}	O_{14}	O_{10}	I_6	O_2
O_{49}	45	41	37	33	29	25	21	O_{17}	O_{13}	O_9	I_5	O_1
O_{48}	44	40	36	32	28	24	20	O_{16}	O_{12}	O_8	O_4	O_0

Z = Empty

TIME OF DAY WORD SENT TO MULTIPLEXOR

(B Register)

PARITY 51	47	43	39	35	31	27	23	19	15	11	7	3
O_{50}	46	42	38	34	30	26	22	18	14	10	6	2
O_{49}	45	41	37	33	29	25	21	17	13	9	5	1
O_{48}	44	40	36	32	28	24	20	16	12	8	4	0

TIME OF DAY BINARY

(Expressed in multiples of 2.4 microseconds.)

TAG = 0

SET INTERRUPT MASK FUNCTION WORD

(A Register)

PARITY												
51	47	43	39	35	31	27	23	O_{19}	O_{15}	O_{11}	I_7	Z_3
O_{50}	46	42	38	34	30	26	22	O_{18}	O_{14}	O_{10}	O_6	Z_2
O_{49}	45	41	37	33	29	25	21	O_{17}	O_{13}	O_9	O_5	Z_1
O_{48}	44	40	36	32	28	24	20	O_{16}	O_{12}	O_8	O_4	I_0

Z = 001, MPX A

= 010, MPX B

= 100, MPX C

INTERRUPT MASK WORD SENT TO MULTIPLEXOR

(B Register)

PARITY												
51	47	43	39	35	31	27	23	19	15	11	7	3
O_{50}	46	42	38	34	30	26	22	18	14	10	6	2
O_{49}	45	41	37	33	29	25	21	17	13	9	MASK 5	1
O_{48}	44	40	36	32	28	24	20	16	12	8	4	0

9 => 0 = Mask field

Bit 0 = Status Change
Bit 1 = DCP 1
Bit 2 = DCP 2
Bit 3 = DCP 3
Bit 4 = DCP 4
Bit 9 = Multiplexor

} 1 = interrupt

TAG = 0

SET GENERAL CONTROL ADAPTER FUNCTION WORD

(A Register)

PARITY 51	47	43	39	35	31	27	23	I_{19}	O_{15}	O_{11}	I_7	O_3
O_{50}	46	42	38	34	30	26	22	O_{18}	O_{14}	N_{10}	O_6	Z_2
O_{49}	45	41	37	33	29	25	21	O_{17}	O_{13}	N_9	I_5	Z_1
O_{48}	44	40	36	32	28	24	20	O_{16}	O_{12}	O_8	O_4	I_0

N = 00, Output Register

N = 01, Interrupt Mask Register

N = 10, Interrupt Register

Z = 01, MPX A

Z = 10, MPX B

(B Register)

The B register contains the output, interrupt mask, or interrupt word to be placed in the GCA register.

TAG = 1

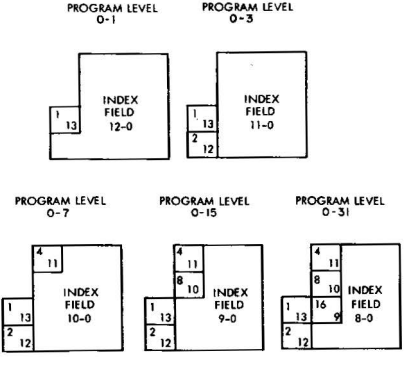
INDIRECT REFERENCE WORD

PARITY												
51	47	43	39	35	31	27	23	19	15	11	7	3
O_{50}	O_{46}	42	38	34	30	26	22	18	14	10	6	2
O_{49}	45	41	37	33	29	25	21	17	ADDRESS 13	9	5	1
I_{48}	44	40	36	32	28	24	20	16	COUPLE 12	8	4	0

Tag - 1

46 - Environment bit: Bit 46 = 0 indicates an IRW. (46 = 1 indicates SIRW)

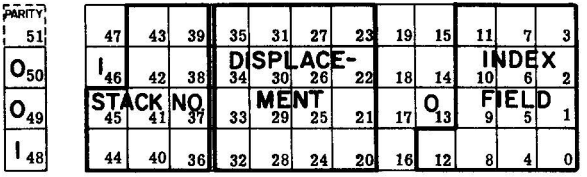
13 = 0 - Address couple: Selects D register (according to current program level as indicated by rLL) and provides index value (see below)



NOTE: The bit order of the LL field is inverted

TAG = 1

STUFFED INDIRECT REFERENCE WORD



Tag - 1

46 - Environment bit: Bit 46 = 1 indicates an SIRW. (46 = 0 indicates IRW)

45 => 36 - Stack number: The number of the stack containing the referenced word

35 => 20 - Displacement field: This number, added to the stack base address, addresses a MSCW.

13 - Always 0 for SIRW

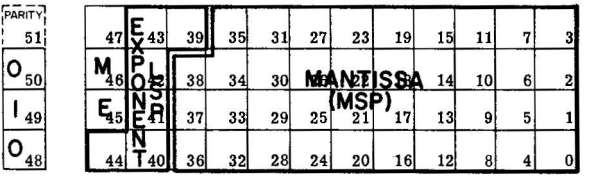
12 => 0 - Index field: This number, added to the address of the MSCW, addresses the referenced word.

(Absolute address of the referenced word = BOSR (as specified by STACK NO) + DISPLACEMENT + INDEX FIELD)

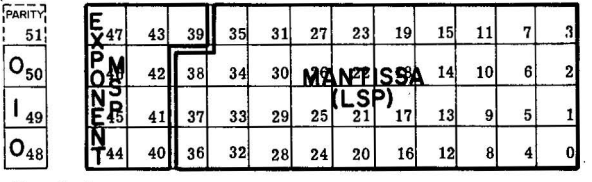
TAG = 2

DOUBLE PRECISION OPERAND

First Word



Second Word



First Word

Tag - 2

M - Sign of mantissa - 1 = negative, 0 = positive

E - Sign of exponent - 1 = negative, 0 = positive

44 => 39 - Exponent, least significant portion

39 => 0 - Mantissa, most significant portion

Second Word

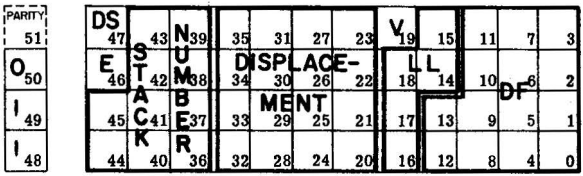
Tag - 2

47 => 39 - Exponent, most significant

38 => 0 - Mantissa, least significant portion

TAG = 3

MARK STACK CONTROL WORD



Tag - 3

DS - Different-stack bit: 1 = different stack, 0 = current stack

E - Environment bit: 1 = active MSCW, 0 = inactive MSCW

45 => 36 - Stack-number field (number of the stack from which PCW was obtained)

35 => 20 - Displacement field - locates MSCW at prior lexicographical level

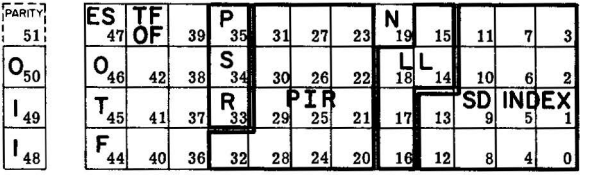
V - Value bit: 1 = operation must continue, 0 = operation to be restarted

LL - Lexicographical level of procedure being entered

DF - Stack history field - locates preceding MSCW (previous "F" register setting)

TAG = 3

RETURN CONTROL WORD



Tag - 3

ES - External sign flip-flop

O - Overflow flip-flop

T - True/false flip-flop

F - Float flip-flop

TFOF - True/false flip-flop occupied flip-flop

PSR - Program syllable (0 thru 5)

32 => 20 - PIR - Index to the program base register

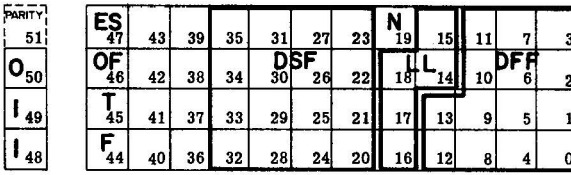
N - 1 = control state, 0 = normal state

LL - The lexicographical level of the calling procedure (at procedure entry)

13 => 0 - SD INDEX - segment descriptor index - If bit 13 = 0, add value specified by bits 12:13 to D[0]; if bit 13 = 1, add value specified by bits 12:13 to D[1].

TAG = 3

TOP OF STACK CONTROL WORD



Tag - 3

ES - External sign flip-flop

OF - Overflow flip-flop

T - True/false flip-flop

F - Float flip-flop

DSF - Delta S-register field (value of rS displacement above BOSR)

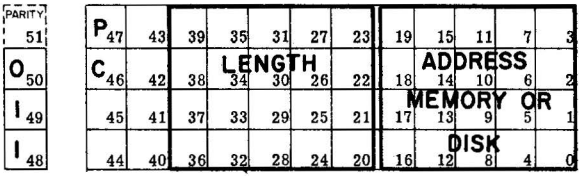
N - Normal-control state flip-flop; 0 = normal

LL - Lexicographic level

DFF - Delta F-register field (value of rF displacement below rS)

TAG = 3

SEGMENT DESCRIPTOR



Tag - 3

P - Presence bit: 1 = present, 0 = absent

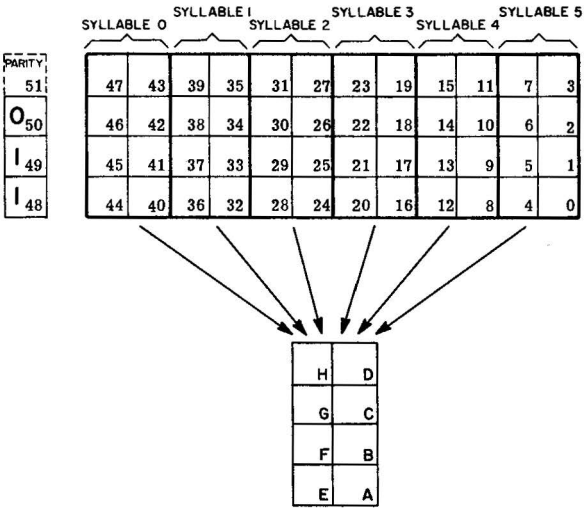
C - Copy bit - 1 = copy, 0 = MOM (original)

39 = 20 - Length of program segment (in words)

19 = 20 - (bit 47 = 1) main memory address, or
(bit 47 = 0 and bit 46 = 0) disk address, non-present segment, or
(bit 47 = 0 and bit 46 = 1) absolute memory address, original program segment descriptor

TAG = 3

PROGRAM WORD (CODE)



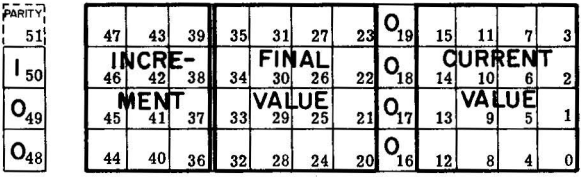
Syllable Format
(Syllable 0-5, depending on PSR Contents)

Syllable Decode Table

(BITS H & G) Identification	Syllable Type	# of Syllables	Function
00	Value Call	2	Brings an operand into the stack.
01	Name Call	2	Brings an IRW into the stack.
1X	Other Operators	1 => 12	Performs the specified operation.

TAG = 4

STEP INDEX WORD



Tag 5

47 => 36 - Increment: Value to be added to current value field.

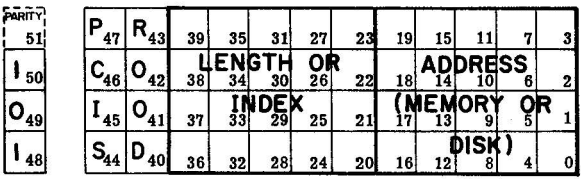
35 => 20 - Final value: Value used to terminate the iteration loop.

19 => 16 - Must be 0 for SIW.

15 => 0 - Current value or count.

TAG = 5

DATA DESCRIPTOR



Tag - 5

P - Presence bit: 1 = present, 0 = absent

C - Copy bit: 1 = copy, 0 = MOM (original)

I - Index bit: 1 = indexing has taken place, 0 = indexing required

S - Segmented bit: 1 = segmented data, 0 = non-segmented data (segment size = 256 words)

R - Read-only-bit: 1 = read-only, 0 = read or write

42 => 41 - Must equal zero

D - Double precision bit: 1 = double, 0 = single precision

39 => 20 - (bit 45 = 0) length of memory area, or
(bit 45 = 1) index value

19 => 0 - (bit 47 = 1) main memory address, or
(bit 47 = 0) disk address

TAG = 5

STRING DESCRIPTOR (NON-INDEXED)

PARITY 51	P 47	R 43	39	35	31	27	23	19	15	11	7	3
I 50	C 46	S 42	38	LENGTH 34	30	26	22	18	ADDRESS 14	10	6	2
O 49	I 45	Z 41	37	33	29	25	21	17	(MEMORY OR 13	9	5	1
I 48	S 44	40	36	32	28	24	20	16	DISK) 12	8	4	0

STRING DESCRIPTOR (INDEXED)

PARITY 51	P 47	R 43	BIN 39	35	31	27	23	19	15	11	7	3
I 50	C 46	S 42	Y 38	34	WORD 30	26	22	18	ADDRESS 14	10	6	2
O 49	I 45	Z 41	TE 37	33	INDEX 29	25	21	17	(MEMORY OR 13	9	5	1
I 48	S 44	40	EX 36	32	28	24	20	16	DISK) 12	8	4	0

Tag - 5

P - Presence bit: 1 = present, 0 = absent

C - Copy bit: 1 = copy, 0 = MOM (original)

I - Index bit: 1 = indexing has taken place, 0 = indexing required

S - Segmented bit: 1 = segmented data, 0 = non-segmented data

R - Read-only bit: 1 = read only, 0 = read or write

SZ - Character size: 100 = 8-bit, 011 = 6-bit, 010 = 4-bit

39 => 20 - Length of memory area (bit 45 = 0), or

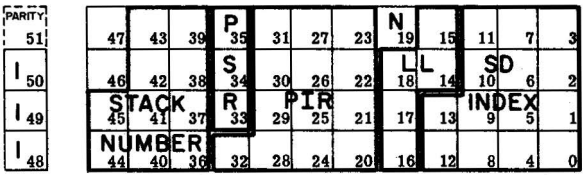
39 => 36 - Byte index (bit 45 = 1)

35 => 20 - Word index (bit 45 = 1)

19 => 0 - (bit 47 = 1) main memory address, or
(bit 47 = 0) disk address

TAG = 7

PROGRAM CONTROL WORD



Tag - 7

45 => 36 - Stack number containing PCW

PSR - Program syllable (0 thru 5)

32 => 20 - PIR - index to the program base register

N - 1 = control, 0 normal state

LL - The lexicographical level of the procedure being entered

13 => 0 - SD INDEX: segment descriptor index - If bit 13 = 0, add value specified by bits 12:13 to D[0] (MCP Code). If bit 13 = 1, add value specified by bits 12:13 to D[1] (User code).

SECTION 5

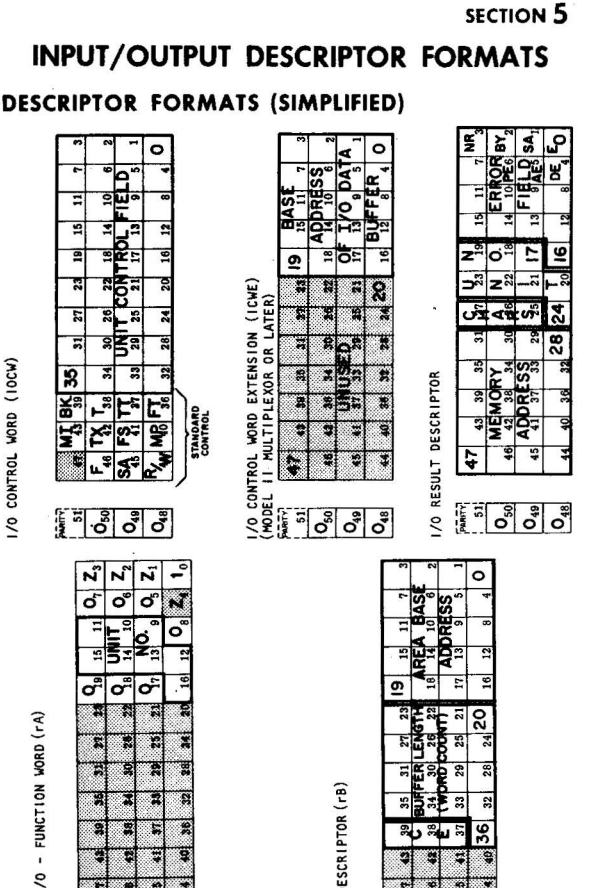
SECTION 5 — CONTENTS

The terms "Multiplexor" and "I/O Processor" are synonymous.

SECTION 5

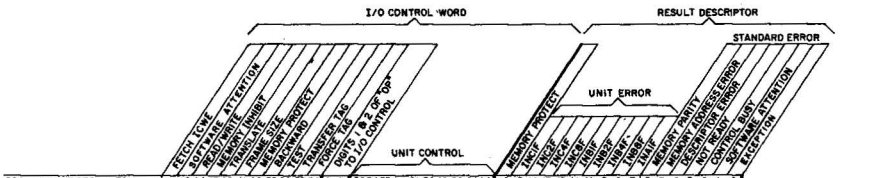
INPUT/OUTPUT DESCRIPTOR FORMATS

DESCRIPTOR FORMATS (SIMPLIFIED)



CARD PUNCH

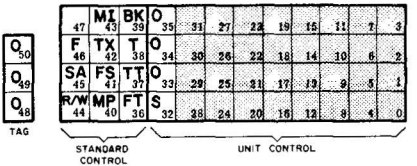
(TYPE = 11)



IOCW/RESULT DESC. BITS	46 45 44 43 42 41 40 39 38 37 36	1	N	35 34 33 32 31 30 29 28 27 26	16 15 14 13 12 11 10 9 8 7	6 5 4 3 2 1 0	
MULTIPLEXOR BITS	51 86 85 84 83 82 81	60 59	A	B	55 87 86 85 84 83 82 77 76 75	52 53 56 74 54 51	UNIQUE RESULT DESCRIPTORS
PERIPHERAL CONTROL BITS					12 11 10 9 8 7 6 5 4	3	
PUNCH BINARY (6-bit from 6-bit)	0 0		2	4	0 0 0 S[1]		
PUNCH EBCDIC (from EBCDIC)	0 1		2	5	0 0 0 S		
PUNCH BCL (from INT. BCL)	1 0		2	3	0 0 0 S		
PUNCH BCL (from EBCDIC)	1 1		2	3	0 0 0 S		
TEST	1		9	9			
					TY	ϕ[2]	GENERAL RESULT DESCRIPTORS
					PC		Punch Check
					MA	MA	Memory Access Error

Notes:
1. S: Stacker
   0 = Primary
   1 = Auxiliary
2. TY = Type
   0 = Model I
   1 = Model II

IOCW



46 => 36 Standard Control Field

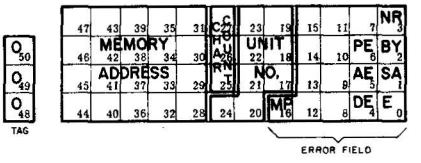
35 => 0 Unit Control Field

32 : Stacker

0 = Primary

1 = Auxiliary

Result Descriptor



47 => 28 Memory Address

27 => 25 Character Count (in binary)

24 => 17 Unit Number

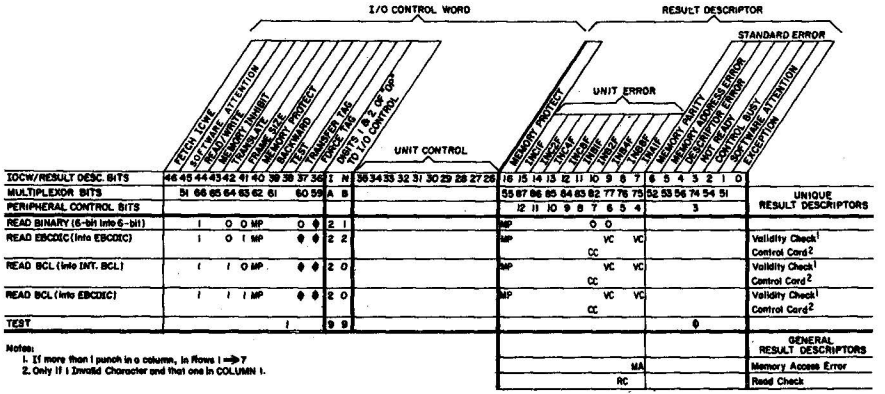
16 => 0 Error Field

0081 = Punch Check

02C1 = Memory Access or Parity Error

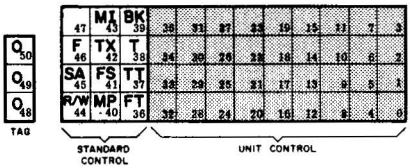
CARD READER

(TYPE = 9)



Notes:
1. If more than 1 punch in a column, in flows 1 ⇒ 7
2. Only if 1 Invalid Character and that one in COLUMN 1.

IOCW



46 => 36 Standard Control Field

42 => 41

0 0 = READ BINARY (6-bit into 6-bit)

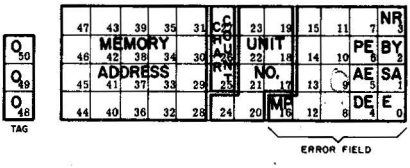
0 1 = READ EBCDIC (into EBCDIC)

1 0 = READ BCL (into INT. BCL)

1 1 = READ BCL (into EBCDIC)

35 => 0 Unit Control Field

Result Descriptor



47 => 28 Memory Address

27 => 25 Character Count (in binary)

24 => 17 Unit Number

16 => 0 Error Field

0081 = Memory Access Error

0101 = Read Check

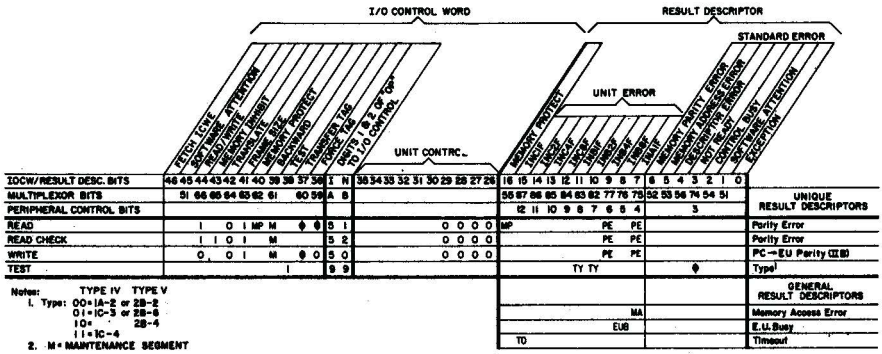
0281 = Validity Check

0381 = Read Check & Validity Check

0401 = Control Card

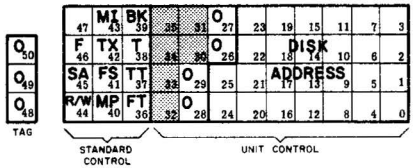
DISK FILE

(TYPE = 1, 3, 33 or 35)



Notes:
1. Type: 00 = IA-2 or 2B-2
01 = IC-3 or 2B-3
10 = 2B-4
11 = IC-4
2. M = MAINTENANCE SEGMENT

IOCW

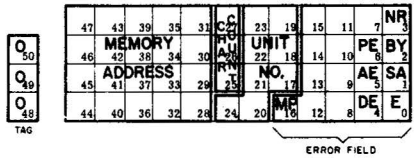


46 => 36 Standard Control Field

35 => 0 Unit Control Field

25 => 0 = Disk Address (in BCD)

Result Descriptor



47 => 28 Memory Address

27 => 25 Character Count (in binary)

24 => 17 Unit Number

16 => 0 Error Field

0081 = Memory Access Error

0101 = E.U. Busy

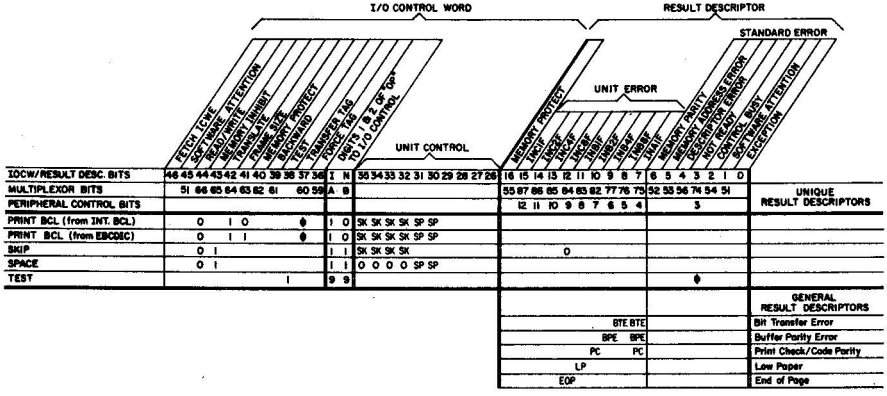
0201 = Write Lockout

0281 = Parity Error

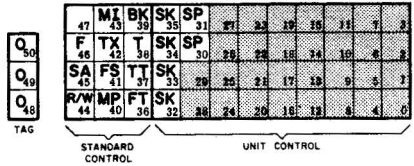
8001 = Timeout

LINE PRINTER

(TYPE = 6, 7, 38 or 39)



IOCW



46 => 36 Standard Control Field

35 => 0 Unit Control Field

35 => 32 : Channel to SKIP to (0 → 11 only)

31 => 30 : Spacing

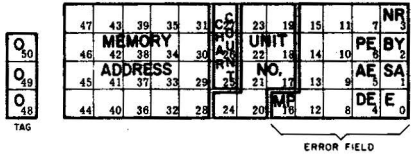
0 0 = No Space

0 1 = Single Space

1 0 = Double Space

1 1 = Double Space

Result Descriptor



47 => 28 Memory Address

27 => 25 Character Count (in binary)

24 => 17 Unit Number

16 => 0 Error Field

0181 = Bit Transfer Error

0281 = Buffer Parity Error

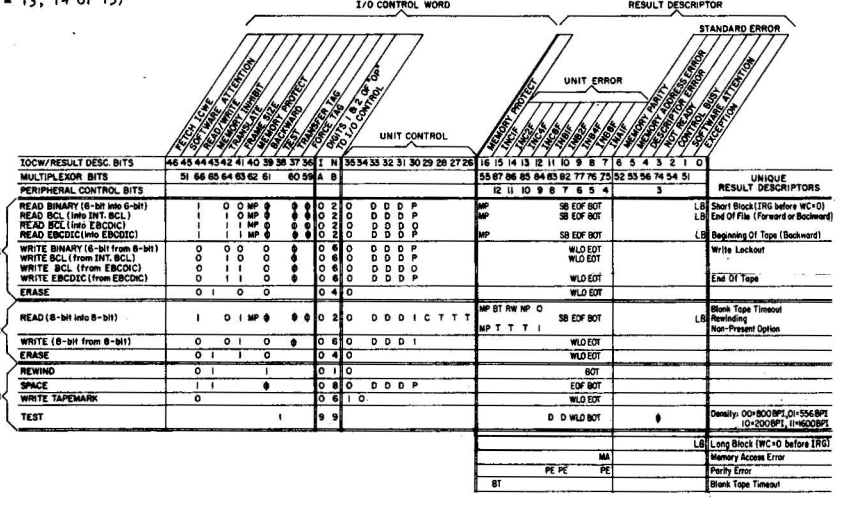
0481 = Print Check/Code Parity

0801 = Low Paper

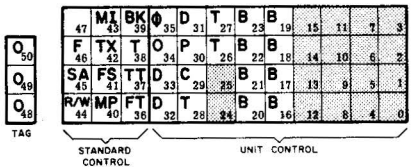
1001 = End of Page

MAG TAPE

(TYPE = 13, 14 or 15)



IOCW



46 => 36 Standard Control Field

35 => 0 Unit Control Field

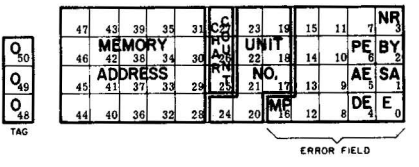
33 32 31 30 : Density, Parity (Peripheral Controls)

0 0 0 0 = Unit Selected, Even
0 0 0 1 = Unit Selected, Odd
1 0 0 0 = 800 BPI, Even (7-Track only; II)
1 0 0 1 = 800 BPI, Odd (7-or 9-Track; II or IV)
1 0 1 0 = 556 BPI, Even
1 0 1 1 = 556 BPI, Odd (7-Track only; II)
1 1 0 0 = 200 BPI, Even
1 1 0 1 = 200 BPI, Odd (7- or 9-Track; II or IV)
1 1 1 0 = None
1 1 1 1 = 1600 BPI, Odd (9-Track P.E. only; V)

28 → 26 = CRC: Track in error, or to correct

23 → 16 = Blocks to Space (in BCD)

Result Descriptor



47 => 28 Memory Address

27 => 25 Character Count (in binary)

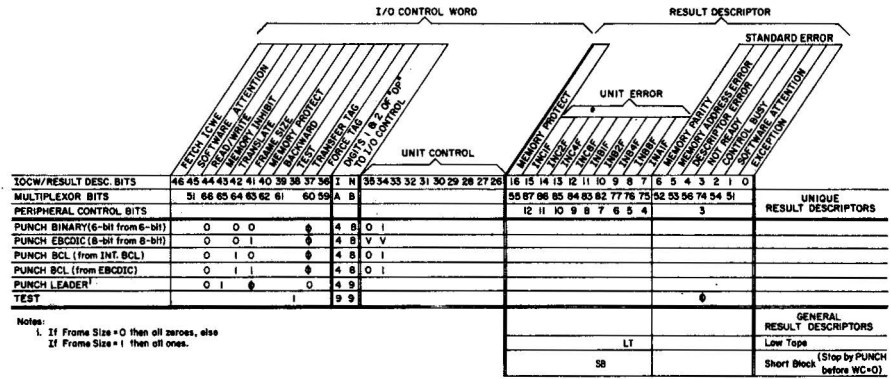
24 => 17 Unit Number

16 => 0 Error Field

0001 = Long Block
0081 = Memory Access Error
0101 = End of Tape or Beginning of Tape
0201 = Write Lockout or End of File
0401 = Short Block
0C81 = Parity Error
0D81 = Parity Error & End Of Tape
0E81 = End Of File & Parity Error
2001 = Non-Present Option
4001 = Rewinding
8001 = Blank Tape Timeout
8101 = Beginning Of Tape & Blank Tape Timeout

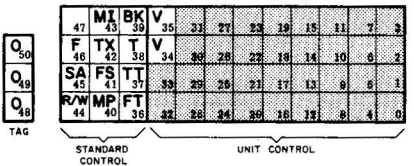
PAPER TAPE PUNCH

(TYPE = 5)



Notes:
1. If Frame Size = 0 then all zeroes, else
   If Frame Size = 1 then all ones.

IOCW



46 => 36 Standard Control Field

35 => 0 Unit Control Field

35 => 34: Punch Character Size, Parity

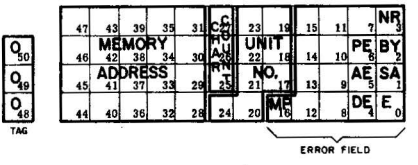
0 0 = 7 Info bits & 1 Parity bit, Even

0 1 = 6 Info bits & 1 Parity bit, Odd
(Stop Codes & Paper Tape Translator are enabled)

1 0 = 8 Info bits, No parity

1 1 = Invalid

Result Descriptor



47 => 28 Memory Address

27 => 25 Character Count (in binary)

24 => 17 Unit Number

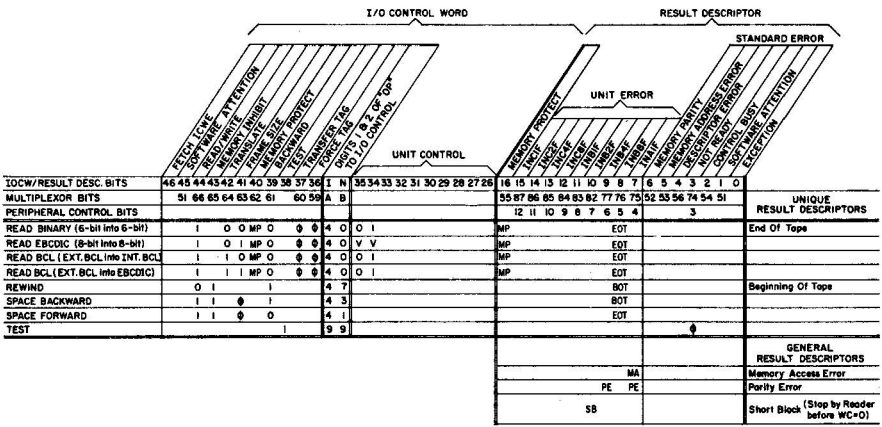
16 => 0 Error Field

0101 = Low Tape

0401 = Short Block

PAPER TAPE READER

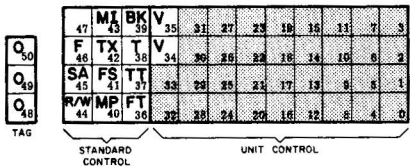
(TYPE = 4)



IOCW/RESULT DESC. BITS	46 45 44 43 42 41 40 39 38 37 36	I	N	35 34 33 32 31 30 29 28 27 26	16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	
MULTIPLEXOR BITS	51 66 65 64 63 62 61 60 59	A	B		55 87 86 85 84 83 82 77 76 75	52 53 56 74 54 51
PERIPHERAL CONTROL BITS					12 11 10 9 8 7 6 5 4	3
READ BINARY (6-bit into 6-bit)	1 0 0 MP 0 ϕ ϕ	4	0	0 1	MP EOT	
READ EBCDIC (8-bit into 8-bit)	1 0 1 MP 0 ϕ ϕ	4	0	V V	MP EOT	
READ BCL (EXT. BCL into INT. BCL)	1 1 0 MP 0 ϕ ϕ	4	0	0 1	MP EOT	
READ BCL (EXT. BCL into EBCDIC)	1 1 1 MP 0 ϕ ϕ	4	0	0 1	MP EOT	
REWIND	0 1 1	4	7		BOT	
SPACE BACKWARD	1 1 ϕ 1	4	3		BOT	
SPACE FORWARD	1 1 ϕ 0 1	4	1		EOT	
TEST	1	9	9			ϕ

GENERAL RESULT DESCRIPTORS	
MA	Memory Access Error
PE PE	Parity Error
SB	Short Block (Stop by Reader before WC=0)

IOCW



46 => 36 Standard Control Field

35 => 0 Unit Control Field

35 => 34: Reader Character Size, Parity

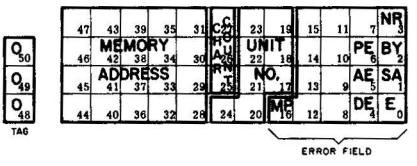
0 0 = 7 Info bits (Bit 8:=0), Even

0 1 = 6 Info bits, Odd
(Stop Codes & Paper Tape Translator are enabled)

1 0 = 8 Info bits, No Parity

1 1 = Invalid

Result Descriptor



47 => 28 Memory Address

27 => 25 Character Count (in binary)

24 => 17 Unit Number

16 => 0 Error Field

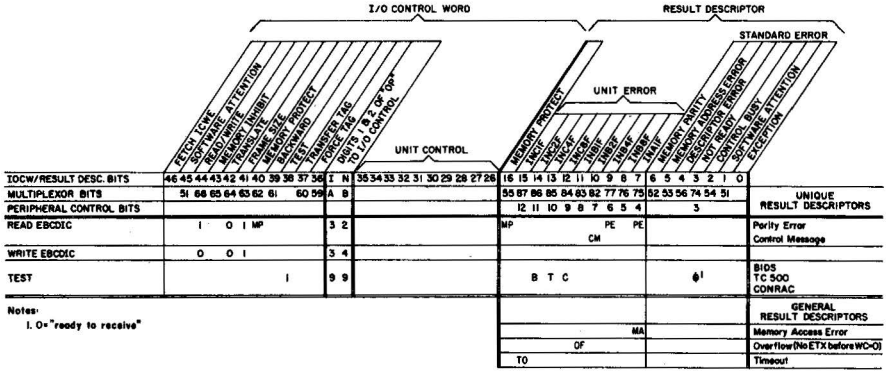
0081 = Memory Access Error

0281 = Parity Error

0401 = Short Block

SINGLE LINE CONTROL

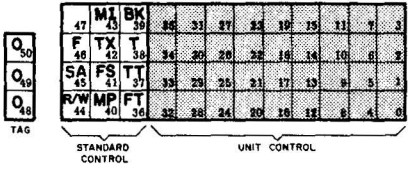
(TYPE =2)



Notes:

1. 0="ready to receive"

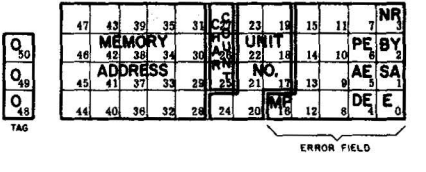
IOCW



46 => 36 Standard Control Field

35 => 0 Unit Control Field

Result Descriptor



47 => 28 Memory Address

27 => 25 Character Count (in binary)

24 => 17 Unit Number

16 => 0 Error Field

0009 = Not Ready to Receive

0081 = Memory Access Error

0281 = Parity Error

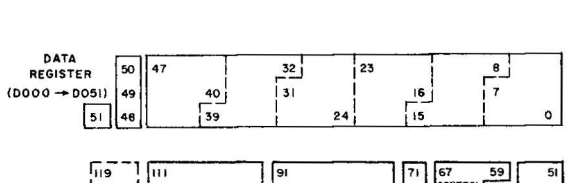
0401 = Control Message

0801 = Overflow

8001 = Timeout

CDR FORMAT

MULTIPLEXOR COMMAND DATA REGISTER FORMAT



Command Register

C051 thru C058 = Error field
- 51 = Attention flag, set by bit 45 of I/O control word.
- 52 = Memory parity error if true.
- 53 = Memory address error if true.
- 54 = Channel busy when true.
- 55 = Memory protect error when true.
- 56 = Descriptor error when true.
- 57 = Card read validity check when true.
- 58 = Set to indicate service cycle output time.

C059 thru C067 = Control/UD field
- 59 & 60 = Tag control bits. (see note)
- 61 = Forward/reverse bit. True for reverse operation.
- 62 = Memory protected write takes place when true.
- 63 = Frame length. True for 8 bit; false for 6 bit.
- 64 = Translate bit. True if MPX translator is used. False when translator is bypassed.
- 65 = True for memory access inhibited.
- 66 = I/O bit. True for input (Read).
- 67 = When true, indicates information transfer is complete during I/O operations.

C068 thru C071 = Four bit character counter (bit 68 is extension bit).

C072 thru C091 = Length of buffer area/result descriptor.

C092 thru C111 = Memory address.

C112 thru C119 = Unit designate.

NOTE: Bits 59 and 60 are set by bits 36 and 37 of the I/O control word. They have the following meaning -

59	60	
0	0	= Store single precision tags.
1	1	= Store double precision tags.
1	0	= Store program tags.
0	1	= Tag field transfer.

Multiplexor Command Register Character Counter

	TAG BIT TIME			
CCCn	71	70	69	68*
0	0	0	0	0
0	0	0	0	1
1	0	0	1	ϕ
2	0	1	1	ϕ
3	0	1	0	ϕ
4	1	1	0	ϕ
5	1	1	1	ϕ
6	1	0	1	ϕ
7	1	0	0	ϕ

*During counts 1 thru 7 C068=0 for forward operation C068=1 for reverse operations ϕ = Indicates a 0 or 1 state.

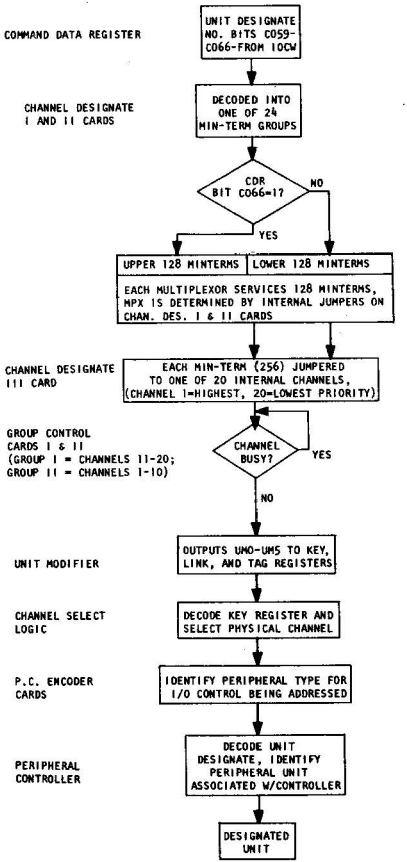
Data Register

D051 = Parity bit.*
D050 thru D048 = Tag bits.
D047 thru D000 = Information bits in either 6 bit or 8 bit character lengths.
*Parity bit does not go to SPM.

UNIT MODIFIER BITS

MPX INTERNAL CHANNEL NO.	UNIT MODIFIER OUTPUT					OCTAL DISPLAY IN TAG & KEY
	UM05	UM04	UM03	UM02	UM01	
1	0	0	1	1	1	07
2	0	0	1	1	0	06
3	0	0	1	0	1	05
4	0	0	1	0	0	04
5	0	1	0	1	1	13
6	0	1	0	1	0	12
7	0	1	0	0	1	11
8	0	1	0	0	0	10
9	1	0	1	1	1	27
10	1	0	1	1	0	26
11	0	1	1	1	1	17
12	0	1	1	1	0	16
13	0	1	1	0	1	15
14	0	1	1	0	0	14
15	1	0	0	1	1	23
16	1	0	0	1	0	22
17	1	0	0	0	1	21
18	1	0	0	0	0	20
19	1	0	1	0	1	25
20	1	0	1	0	0	24

UNIT DESIGNATE DECODING



CONFIGURATION INSTRUCTIONS

PCC Cabinet

Enter type of peripheral control in Column A of configuration chart.
Enter MPX internal priority assignment in Column B (1-20).

Channel Assignment

Enter 2 X priority in Column D.
Enter 2 X priority -1 in Column F.
Jumpering on the four cards is taken from Column C to D and Column E to F.

Group Assignment

Enter the group assignment for each control in Column C.
When selecting a unit designate minterm from the min-term group/unit designate table (there are a total of 256 minterms possible), the lower group (left column) is automatically selected if no jumper is installed in the location shown. To switch the selection to a group within the upper 128 minterms, install the appropriate jumper in the plug-ons for the card locations shown in the table.
This table also shows the jumpers necessary to expand an 8 minterm to a 16 minterm.
Jumpering is done on plug-ons provided with the four cards concerned with columns H and I.
Enter from the table provided, the pin assignment per group in Column J.
Enter 2 X priority in Column K (see notes 1 and 2).

Note 1: Multiple group pads may only be used when it is desired to expand two 10 sized groups to a 20 size group. Minterm group 0-9 (128-137) may be combined with groups 16-25 (144-153) or 48-57 (176-185). Minterm group 32-41 (160-169) may be combined with groups 48-57 (176-185) or 16-25 (144-153).

Note 2: If a multi-control sub-system is to be implemented (i.e., controls common to the same exchange), the appropriate group pads (Col J) must be jumpered to each channel pad associated with a channel within the multi-control group.

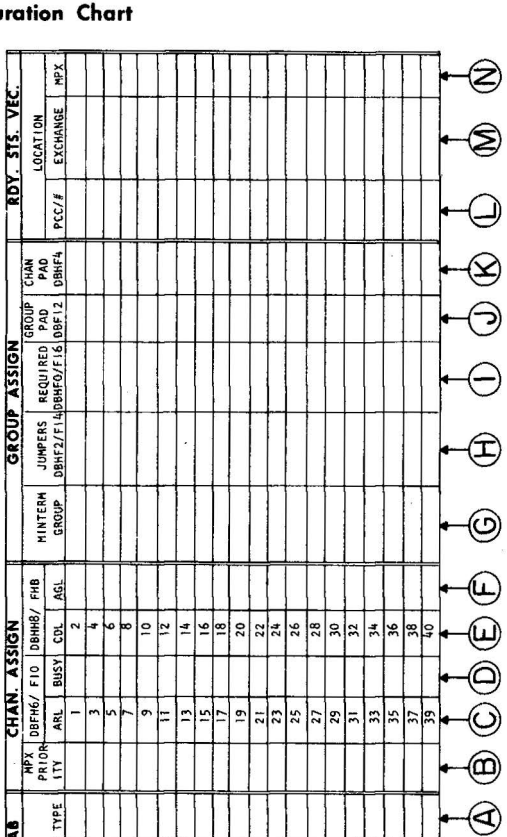
Jumpering on the two cards is taken from Column J to K.

Ready Status

Enter the exchange or PCC locations from PCC and Exchange tables in Column L or M.
Enter from the Vector table the MPX location in Column N.
Use pushpin jumpers on the backplane to connect from Column L or M to N.

MPX CONFIGURATION

Configuration Chart



Min-Term Group/Unit Designate Table

MINTERM GROUPS			MINTERM ASSIGN		UNIT	CHANNEL
			(H)	(I)	(J)	(K)
LOWER 128 MINTERMS	UPPER 128 MINTERMS	GROUP SIZE	UPPER JUMPERS DBHF2/DBF14	DBHF0/DBF16	DECODE DESIGNATE DBFI2	/ DBHF4
0-9	128-137	10	A		1	*
16-25	144-153	10	B		3	*
64-71 or 64-79	192-199 or 192-207	8 or 16	C		5	*
80-87 or 80-95	208-215 or 208-223	8 or 16	D		7	*
10	138	1	H		9	*
11	139	1	J		11	*
12	140	1	K		13	*
13	141	1	L		15	*
14	142	1	M		17	*
15	143	1	N		19	*
32-41	160-169	10		A	21	*
48-57	176-185	10		B	23	*
96-103 or 96-111	224-231 or 224-239	8 or 16		C	25	*
112-119 or 112-127	240-247 or 240-255	8 or 16		D	27	*
26	154	1		H	29	*
27	155	1		J	31	*
28	156	1		K	33	*
29	157	1		L	35	*
30	158	1		M	37	*
31	159	1		N	39	*
42-43	170-171	2	P		41	*
58-59	186-187	2		P	42	*
44-47	172-175	4	E		43	*
60-63	188-191	4		E	44	*

UNIT MINTERM GROUP			
8 SIZED GROUP	16 SIZED GROUP	DBHF2/DBF14	DBHF0/DBF16
64-71	64-79	Y	
192-199	192-207	Y	
80-87	80-95	Z	
208-215	208-223	Z	
96-103	96-111		Y
224-231	224-239		Y
112-119	112-127		Z
240-247	240-255		Z

*Note 1: Multiple group pads may only be used when it is desired to expand two 10 sized groups to a 20 size group. Minterm group 0-9 (128-137) may be combined with groups 16-25 (144-153) or 48-57 (176-185). Minterm group 32-41 (160-169) may be combined with groups 48-57 (176-185) or 16-25 (144-153).

*Note 2: If a multi-control sub-system is to be implemented (i.e., controls common to the same exchange), the appropriate group pads (Col J) must be jumpered to each channel pad associated with a channel within the multi-control group.

MPX CONFIGURATION

PCC, Exchange, and Vector Tables

(L)

PCC-0-DCHB4 / PCC-1-DCHB8	
CON	PIN
0	0D/0I
1	1D/1I
2	0C/0V
3	1C/1V
4	1H/0U
5	0H
6	0G
7	0E
8	1J
9	0J

(M)

EXCHANGE	
#	CARD
1	DCHC2
2	DCHC6
3	DCHD0
4	DCJA0
5	DCJA4
6	DCJA8
7	DCJB2
8	DCJB6
9	DCJC0
10	DCJC4
11	DCJC8

UNIT	PIN	UNIT	PIN
0	0D	10	0I
1	1D	11	1I
2	0C	12	0V
3	1C	13	1V
4	1H	14	0U
5	0H	15	1U
6	0G	16	0X
7	0E	17	1X
8	1J	18	1W
9	0J	19	0W

(N)

VECTOR	0	3	6		1	4	7		2	5
PIN	DCH-LOCATION			PIN	DCH-LOCATION			PIN	DCH-LOC	
	A0	A2	A4		A0	A2	A4		A0	A2
1K	0	96	192	0M	32	128	224	1L	64	160
1X	1			0W				0Y		
0T	2			1W				1V		
1R	3			1T				0S		
1N	4			0P				1P		
0J	5			0L				1M		
0B	6			0E				1H		
0C	7			0F				1I		
	A6	A8	B0		A6	A8	B0		A6	A8
0N	8	104	200	1Q	40	136	232	1E	72	168
1K	9			0M				1L		
1X	10			0W				0Y		
0T	11			1W				1V		
1R	12			1T				0S		
1N	13			0P				1P		
0J	14			0L				1M		
0B	15			0E				1H		
0C	16			0F				1I		
	DCF-LOCATION				DCF-LOCATION				DCF-LOC	
	B2	B4	B6		B2	B4	B6		B2	B4
0N	17	113	209	1Q	49	145	241	1E	81	177
1K	18			0M				1L		
1X	19			0W				0Y		
0T	20			1W				1V		
1R	21			1T				0S		
1N	22			0P				1P		
0J	23			0L				1M		
0B	24			0E				1H		
0C	25			0F				1I		
	B8	C0	C2		B8	C0	C2		B8	C0
0N	26	122	218	1Q	58	154	250	1E	90	186
1K	27			0M				1L		
1X	28			0W				0Y		
0T	29			1W				1V		
1R	30			1T				0S		
1N	31			0P				1P		

Adapter Locations

PRIORITY	ATR KBY LINK (OCTAL)	CARD LOCATION
1	07	DBHF6
2	06	DBJF4
3	05	DBHG8
4	04	DBJG8
5	13	DBHF8
6	12	DBJF6
7	11	DBHGo
8	10	DBJF8
9	27	DBHG2
10	26	DBJGO
11	17	DBHG4
12	16	DBJG2
13	15	DBHHO
14	14	DBJHO
15	23	DBHH2
16	22	DBJH2
17	21	DBHH4
18	20	DBJH4
19	25	DBHH6
20	24	DBJH6

SECTION 6

HARDWARE INTERRUPTS

SECTION 6 – CONTENTS

The terms "Multiplexor" and "I/O Processor" are synonymous.

HARDWARE INTERRUPTS

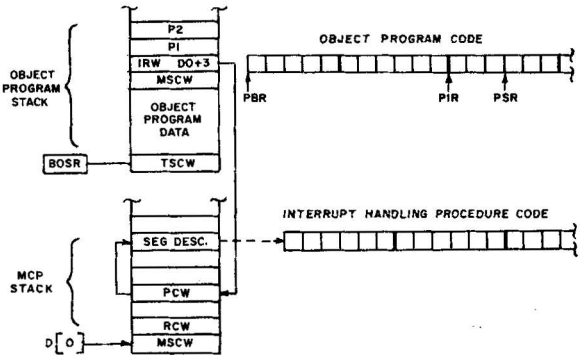
PRESENCE BIT INTERRUPT CHART

Presence Bit Interrupt Condition		P2	$P1_{(3)}$	RT Bit (3) (bit 46)	Returning Operator	PIR,PSR New RCW	Software Function
Data Dependent	Stack Vector Data Descriptor during data reference	(1) IRW (stuffed)	Interrupt ID	0	Exit	S_n (4)	Make stack or stack vector present
		(2) IRW (stuffed)	Interrupt ID	1	Return	S_n (4)	
	Data Descriptor during data reference	(1) Data Descriptor (copy)	Interrupt ID	0	Exit	S_n (4)	Search stack for copies of not present data descriptor, make original Mom and copies present, return data descriptor where noted.
		(2) Data Descriptor (copy)	Interrupt ID	1	Return	S_n (4)	
Procedure Dependent	Stack Vector or Stack Vector Data Descriptor during display update	- Data Descriptor (copy)	Interrupt ID	0	Exit	From RCW/PCW	
	Segment Descriptor	- Segment Descriptor (copy)	Interrupt ID	0	Exit	From RCW/PCW	Locate segment descriptor (Mom) via copy in P_2; address field of copy points to Mom.

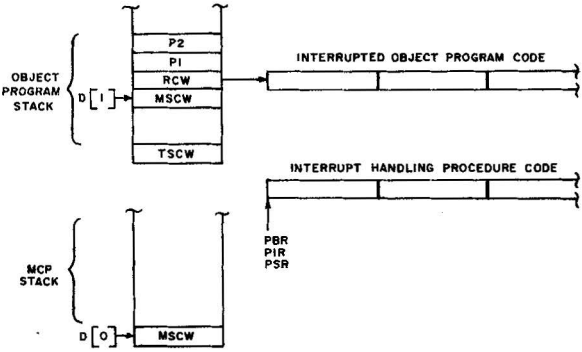
(1) Value Call or Enter
(2) All operators except Value Call, Enter, or Move Stack
(3) RT bit is packed in the int. I.D. (P_1)
(4) S_n indicates the PIR and PSR point to current operator syllable

STACK AT INTERRUPT PROCEDURE

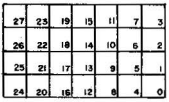
Stack Format Prior To Calling The Interrupt Procedure



Stack Format After Entering The Interrupt Procedure



P1:

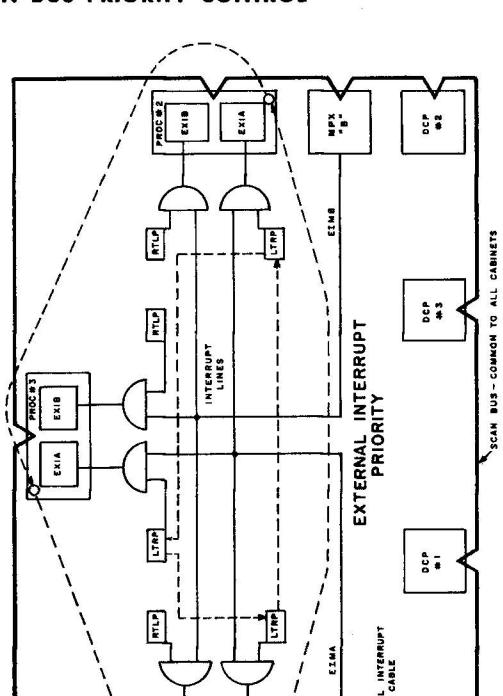


External	0	1	0	0	0	1	n	DCP-1
	0	1	0	0	0	2	n	DCP-2
	0	1	0	0	0	3	n	DCP-3
	0	1	0	0	0	4	n	DCP-4
	0	1	0	0	0	5	n	External MPX
	0	1	0	0	0	9	n	I/O Finish
	0	1	0	0	0	F	n	Status Change
	0	2	0	0	0	0	0	Processor to Processor
Special	0	4	0	0	0	0	1	Interval Timer
	0	4	0	0	0	0	2	Stack Overflow (rS≥LOSR)
	0	8	0	0	0	2	0	Invalid Index Inaccurate
	0	8	0	0	1	0	0	*Presence Bit PIR et al in RCW
	0	8	0	0	2	0	0	Sequence Error (Display update)
	1	0	0	0	0	0	0	Programmed Operator
	1	0	0	0	0	0	1	Memory Protect
	1	0	0	0	0	0	2	Invalid Operation
Syllable Dependent	1	0	0	0	0	0	4	Divide by Zero
	1	0	0	0	0	0	8	Exponent Overflow
	1	0	0	0	0	1	0	Exponent Underflow
	1	0	0	0	0	2	0	Invalid Index
	1	0	0	0	0	4	0	Integer Overflow
	1	0	0	0	0	8	0	Base of Stack (r ≤ BOSR)
	1	0	0	0	1	0	0	*Presence Bit
	1	0	0	0	2	0	0	Sequence Error (Display Update)
	1	0	0	0	4	0	0	**Segmented Array
Alarm	2	0	0	0	0	0	1	Loop (Instruction > 2 sec.)
	2	0	0	0	0	0	2	Memory Parity Error
	2	0	0	0	0	0	4	Scan Parity Error
	2	0	0	0	0	0	8	Invalid Address
	2	0	0	0	0	1	0	Stack Underflow (r S ≤ r F)
	2	0	0	0	0	2	0	Invalid Program Word (Tag ≠ 3)

*Bit 46 (RT bit) may be 0 or 1
[See Presence Bit Interrupt Chart]
**P2 contains 3-bit literal to indicate number of words in stock below MSCW.

HARDWARE INTERRUPTS

SCAN BUS PRIORITY CONTROL



SECTION 7

MISCELLANEOUS HARDWARE INFORMATION

SECTION 7 — CONTENTS

The terms "Multiplexor" and "I/O Processor" are synonymous.

SECTION 7

MISCELLANEOUS HARDWARE INFORMATION

CONSOLE DISPLAY CONTROL TRANSLATOR CHART

8 BIT INA REGISTER BITS	EBCDIC INB REGISTER BITS	7 BIT ASCII CODE WITHIN THE SHIFT REGISTER	CHARACTER
8 4 2 1	8 4 2 1		
1 1 0 0	0 0 0 1	4/1	A
1 1 0 0	0 0 1 0	4/2	B
1 1 0 0	0 0 1 1	4/3	C
1 1 0 0	0 1 0 0	4/4	D
1 1 0 0	0 1 0 1	4/5	E
1 1 0 0	0 1 1 0	4/6	F
1 1 0 0	0 1 1 1	4/7	G
1 1 0 0	1 0 0 0	4/8	H
1 1 0 0	1 0 0 1	4/9	I
1 1 0 1	0 0 0 1	4/10	J
1 1 0 1	0 0 1 0	4/11	K
1 1 0 1	0 0 1 1	4/12	L
1 1 0 1	0 1 0 0	4/13	M
1 1 0 1	0 1 0 1	4/14	N
1 1 0 1	0 1 1 0	4/15	O
1 1 0 1	0 1 1 1	5/0	P
1 1 0 1	1 0 0 0	5/1	Q
1 1 0 1	1 0 0 1	5/2	R
1 1 1 0	0 0 1 0	5/3	S
1 1 1 0	0 0 1 1	5/4	T
1 1 1 0	0 1 0 0	5/5	U
1 1 1 0	0 1 0 1	5/6	V
1 1 1 0	0 1 1 0	5/7	W
1 1 1 0	0 1 1 1	5/8	X
1 1 1 0	1 0 0 0	5/9	Y
1 1 1 0	1 0 0 1	5/10	Z
1 1 1 1	0 0 0 0	3/0	0
1 1 1 1	0 0 0 1	3/1	1
1 1 1 1	0 0 1 0	3/2	2
1 1 1 1	0 0 1 1	3/3	3
1 1 1 1	0 1 0 0	3/4	4
1 1 1 1	0 1 0 1	3/5	5
1 1 1 1	0 1 1 0	3/6	6
1 1 1 1	0 1 1 1	3/7	7
1 1 1 1	1 0 0 0	3/8	8
1 1 1 1	1 0 0 1	3/9	9
0 1 0 0	1 0 1 0	5/11	[
0 1 0 0	1 0 1 1	2/14	.
0 1 0 0	1 1 0 0	3/12	<
0 1 0 0	1 1 0 1	2/8	(
0 1 0 0	1 1 1 0	2/11	+
0 1 0 0	1 1 1 1	2/1	≥
0 1 0 1	0 0 0 0	2/6	&
0 1 0 1	1 0 1 0	5/13	]

CONSOLE DISPLAY CONTROL

CONSOLE DISPLAY CONTROL TRANSLATOR CHART (Cont)

8 BIT INA REGISTER BITS	EBCDIC INB REGISTER BITS	7 BIT ASCII CODE WITHIN THE SHIFT REGISTER	CHARACTER
8 4 2 1	8 4 2 1		
0 1 0 1	1 0 1 1	2/4	$
0 1 0 1	1 1 0 0	2/10	*
0 1 0 1	1 1 0 1	2/9	)
0 1 0 1	1 1 1 0	3/11	;
0 1 0 1	1 1 1 1	5/14	¢
0 1 1 0	0 0 0 0	2/13	-
0 1 1 0	0 0 0 1	2/15	/
0 1 1 0	1 0 1 1	2/12	,
0 1 1 0	1 1 0 0	2/5	%
0 1 1 0	1 1 0 1	5/15	_
0 1 1 0	1 1 1 0	3/14	>
0 1 1 0	1 1 1 1	3/15	?
0 1 1 1	1 0 1 0	3/16	:
0 1 1 1	1 0 1 1	2/3	#
0 1 1 1	1 1 0 0	4/10	@
0 1 1 1	1 1 0 1	2/7	≤
0 1 1 1	1 1 1 0	3/13	=
0 0 0 0	0 0 0 1	0/1	SOH (START OF HEADING)
0 0 0 0	0 0 1 0	0/2	STX (START OF TEXT)
0 0 0 0	0 0 1 1	0/3	ETX (END OF TEXT)
0 0 0 0	0 1 0 0	0/4	EOT (END OF TRANSMISSION)
0 0 0 0	0 1 0 1	0/5	ENQ (ENQUIRY)
0 0 0 0	0 1 1 0	0/6	ACK (ACKNOWLEDGE)
0 0 0 0	1 0 0 0	0/8	BS (BACK SPACE)
0 0 0 0	1 0 0 1	0/9	TAB (HORIZONTAL)
0 0 0 0	1 0 1 0	0/10	LF (LINE FEED)
0 0 0 0	1 0 1 1	0/11	SET TAB
0 0 0 0	1 1 0 0	0/12	FF (FORM FEED-MASTER CLEAR)
0 0 0 0	1 1 0 1	0/13	CR ▽ (CARRIAGE RETURN)
0 0 0 0	1 1 1 0	0/14	SO◁ (SHIFT OUT)
0 0 0 0	1 1 1 1	0/15	SI▷ (SHIFT IN)
0 0 0 1	0 0 0 1	1/1	LINE ERASE
0 0 0 1	0 0 1 0	1/2	FS (FORWARD SPACE)
0 0 0 1	0 0 1 1	1/3	RLF (REVERSE LINE FEED)
0 0 0 1	0 1 0 0	1/4	HOME
0 0 0 1	0 1 0 1	1/5	NAK (NEGATIVE ACKNOWLEDGE)

CONSOLE DISPLAY CONTROL TRANSLATOR CHART (Cont)

8 BIT INA REGISTER BITS	EBCDIC INB REGISTER BITS	7 BIT ASCII CODE WITHIN THE SHIFT REGISTER	CHARACTER
8 4 2 1	8 4 2 1		
0 0 0 1	1 1 0 1	1/13	GSΔ(GROUP SEPARATOR)
0 0 0 1	1 1 1 0	1/14	*RS∇(RECORD SEPARATOR)
0 0 0 1	1 1 1 1	1/15	*US $\triangleright$ (UNIT SEPARATOR)
*RS replaces SO in Phase II Console Display Control. US replaces SI in Phase II Console Display Control.			

DISK FILES

Storage Unit Characteristics

TYPE	AVG ACCESS (Millisec)	Avg. Trans. Rates in bytes/sec.	RPM	BITS CHAR	CHARS SEG	SEGS SU	SEGS FACE	TRACKS ZONE	SEGS TRACK	(30-Word Segs)/Track in Each Zone Z1	Z2	Z3	Z4	Z5	Z6	Z7	DISKS SU
IA2	20	312K	1500	8	180	60400	7550	50	151	43	54	54					4
IC3	25	345K	1200	8	180	111200	13900	50	278	73	95	110					4
IC4	40	215K	750	8	180	111200	13900	50	278	69	89	120					4
IIB2	35	448K	860	8	180	222400	55600	50	1112	103	122	139	158	177	197	216	2
IIB4	60	263K	500	8	180	222400	55600	50	1112	103	122	139	158	177	197	216	2
IIB6	20	671K	1500	8	180	167600	41900	50	838	103	122	139	158	158	158		2

IA and IC:
For addresses starting at 0 and going thru 60399, the order of access to faces/tracks/zones is:

```
F1/T00/Z1 → F1/T00/Z2 → F1/T00/Z3 →
→F1/T01/Z1 → F1/T01/Z2 → F1/T01/Z3 →
                  ⋮
→F1/T49/Z1 → F1/T49/Z2 → F1/T49/Z3 →
*→F4/T00/Z1 ... → F4/T49/Z3 →
→F2/T00/Z1 ... → F2/T49/Z3 →
→F3/T00/Z1 ... → F3/T49/Z3 →
→F5/ ... → F8/ ... → F6/ ... → F/7 ...
```

* For IA2, order of access to faces is F1→F4→F2→F3→F5→F8→F6→F7.
For IC, order of access to faces is F1→F2→F3 ... F8.

IIB
Order-of-access to the zone areas on the 4 faces is indicated by circled numbers for IIB2 and IIB4 units. Absolute disk address of the first segment in the zero track for the zone areas is shown for IIB2/IIB4 in the left-hand "enclosure" and for IIB6 in the right-hand "enclosure".

Order-of-access to faces is F1-F2-F3-F4. The most rapidly varying address component for IIB is track, followed by face, zone area and disk.

Odd-numbered zone areas are "L" zones and contain tracks 00→49; even-numbered "U" zones contain tracks 50→99.

Disk File Lockout Charts

MODEL 1A-2 (20 ms) DISK

SWITCH*	STARTING ADDRESS	ENDING ADDRESS	ADDRESS LOCATED IN
MASTER	000000	030199	
1 and 2	000000	030199	First SU
3 and 4	030200	060399	First SU
5 and 6	060400	090399	Second SU
7 and 8	090400	120799	Second SU
9 and 10	120800	150999	Third SU
11 and 12	151000	181199	Third SU
	NOTE		
	No more than 3 SU's may be connected to an EU; therefore, switches 13 thru 20 are not normally used.		
13 and 14	181200	211399	
15 and 16	211400	241599	
17 and 18	241600	271799	
19 and 20	271800	301999	

* Switches must be used in pairs as shown.

DISK FILE LOCKOUT

MODELS 1C-3 and 1C-4 DISK

SWITCH	STARTING ADDRESS	ENDING ADDRESS	LOCKS OUT
MASTER	000000	555999	
1	000000	027799	1st. DFSU, 1st. disk
2	027800	055999	1st. DFSU, 2nd. disk
3	055600	083399	1st. DFSU, 3rd. disk
4	083400	111199	1st. DFSU, 4th. disk
5	111200	138999	2nd. DFSU, 1st. disk
6	139000	166799	2nd. DFSU, 2nd. disk
7	166800	194599	2nd. DFSU, 3rd. disk
8	194600	222399	2nd. DFSU, 4th. disk
9	222400	250199	3rd. DFSU, 1st. disk
10	250200	277999	3rd. DFSU, 2nd. disk
11	278000	305799	3rd. DFSU, 3rd. disk
12	305800	333599	3rd. DFSU, 4th. disk
13	333600	361399	4th. DFSU, 1st. disk
14	361400	389199	4th. DFSU, 2nd. disk
15	389200	416999	4th. DFSU, 3rd. disk
16	417000	444799	4th. DFSU, 4th. disk
17	444800	472599	5th. DFSU, 1st. disk
18	472600	500399	5th. DFSU, 2nd. disk
19	500400	528199	5th. DFSU, 3rd. disk
20	528200	555999	5th. DFSU, 4th. disk

MODEL 2 (35 ms, 180 character) DISK 2B2

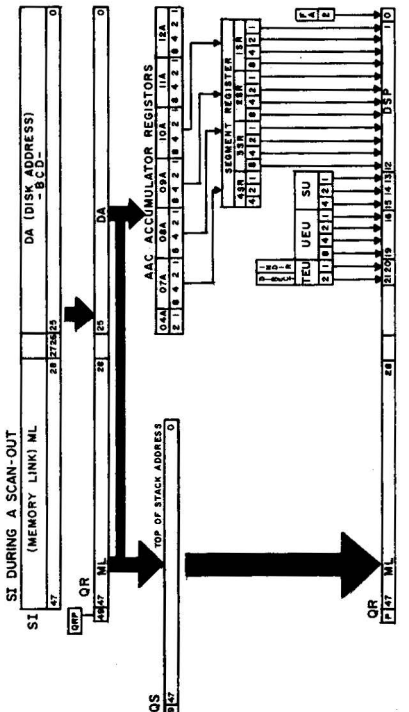
SWITCH	STARTING ADDRESS	ENDING ADDRESS	LOCKS OUT
MASTER	0000000	1111999	Entire Bank
0	0000000	0111199	SU1, Module A
1	0111200	0222399	SU1, Module B
2	0222400	0333599	SU2, Module A
3	0333600	0444799	SU2, Module B
4	0444800	0555999	SU3, Module A
5	0556000	0667199	SU3, Module B
6	0667200	0778399	SU4, Module A
7	0778400	0889599	SU4, Module B
8	0889600	1000799	SU5, Module A
9	1000800	1111999	SU5, Module B

MODEL 2 (20 ms) DISK 2B6

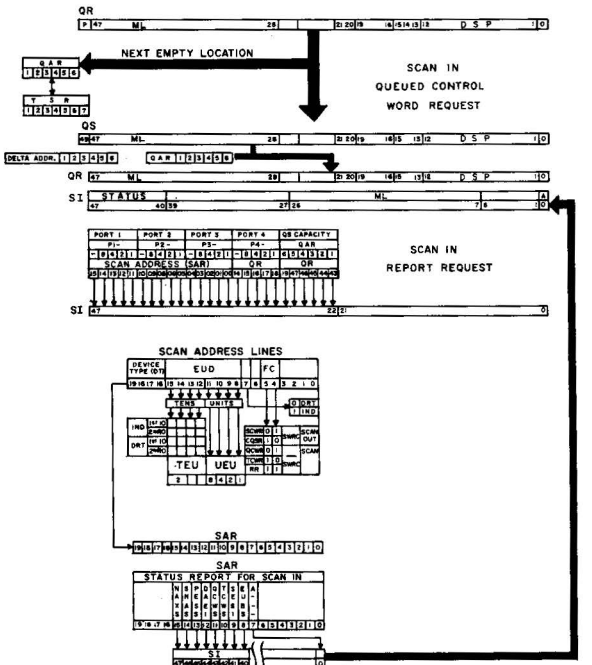
SWITCH	STARTING ADDRESS	ENDING ADDRESS	LOCKS OUT
MASTER	000000	837999	Entire Bank
0	000000	083799	SU1, Module A
1	083800	167599	SU1, Module B
2	167600	251399	SU2, Module A
3	251400	335199	SU2, Module B
4	335200	418999	SU3, Module A
5	419000	502799	SU3, Module B
6	502800	586599	SU4, Module A
7	586600	670399	SU4, Module B
8	670400	754199	SU5, Module A
9	754200	837999	SU5, Module B

DISK FILE OPTIMIZER

Register Flow

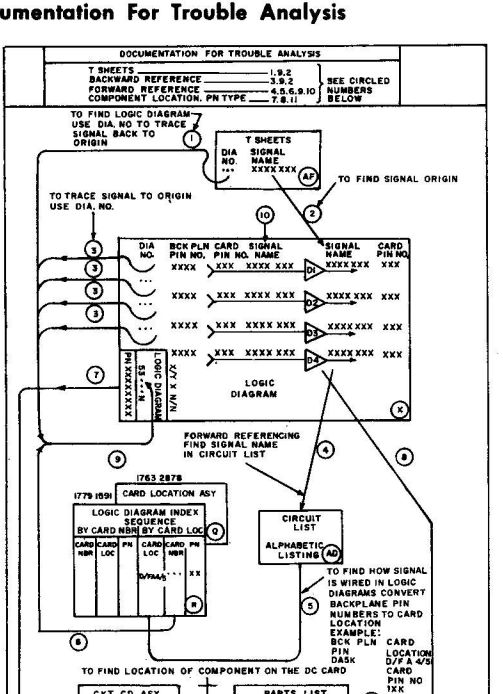


Register Flow (Cont) DISK FILE OPTIMIZER

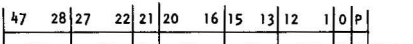


DISK FILE OPTIMIZER

Documentation For Trouble Analysis



DFO Stack Format



BITS 47 => 28 = Memory Link (ML)

27 => 22 = Spare bits - Not used (SP)

21 = Disk Exchange Bit (E)

0 = Disk exchange A (Direct)

1 = Disk exchange B (Indirect EU)

20 => 16 = EU number (EU)

Bit 20 = which group of 10 EU's.

19 => 16 = EU0-9

15 => 13 = SU number (SU)

12 => 1 = Desired Shaft Position, binary (DSP)

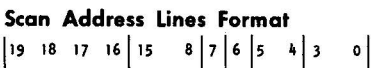
0 = Shaft bit. (S)

Differentials the two shafts of an SU, when applicable.

P = parity bit.

DISK FILE OPTIMIZER

Scan Address Lines Format



19 => 6 = Device Type (DT)

Code 1001 selects the Disk File Optimizers.

15 => 8 = Electronics Unit Designate (EUD)

Defines the exchange and the EU number associated with the job.

7 =

6 = Spare bit, not used (SP).

5 => 4 = Function Code (FC)

Defines the operation requested by the IOM:

Scan out:

01 = store CW. request.

10 = clear the stack request.

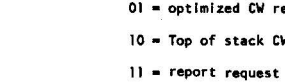
Scan In:

01 = optimized CW request

10 = Top of stack CW request

11 = report request

Scan Out Word Format



47 => 28 = Memory Link (ML)

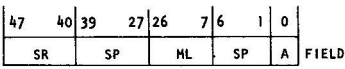
Defines the complete memory link address.

27 => 26 = Spare bits, not used (SP).

25 => 0 = Disk Address (DA)

Defines the 6 BCD characters plus the 2 expansion bits of the desired disk starting address, not including the desired EU, or the desired DEX.

Scan In Word Format



47 = 40 = Status Report (SR)

SR Bit	ML	Level	Status Condition
47	ACW	NAXS	No Access to QEX
46	ACW	SNAS	SU Not Available
45	ACW	PESS	QS Parity Error
44	ACW	DAEI	Disk Address Error
43	ACW	QCWS	Optimized CW
42	ACW	TCWS	Top of the Stack CW
41	Zeros	SESI	Stack Empty
40	Zeros	EUBS	CW Not Available

39 = 27 = Spare bits (SP)

26 = 7 = Memory Link (ML)

Complete Memory Link Address.

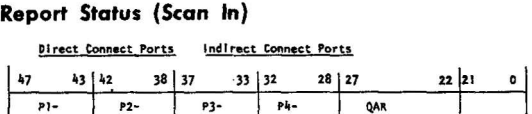
6 = 1 = Spare bits (SP)

0 = Attention field (A)

1 = IOM to examine SR field.

DISK FILE OPTIMIZER

Report Status (Scan In)



47 - 43 = P1-n (Port 1)

42 - 38 = P2-n (Port 2)

37 - 33 = P3-n (Port 3)

32 - 28 = P4-n (Port 4)

Where n = -, 8, 4, 2, or 1. The - bit in each port indicates that the corresponding 8, 4, 2, or 1 bits are valid information representing an EUD code group of ten possible DFEU's.

27 - 22 = Stack capacity of that DFO.

Electronics Unit Designate Group Numbers

EU#	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240	←Lower
1	1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241	Limits
2	2	18	34	50	66	82	98	114	130	146	162	178	194	210	226	242	
3	3	19	35	51	67	83	99	115	131	147	163	179	195	211	227	243	
4	4	20	36	52	68	84	100	116	132	148	164	180	196	212	228	244	
5	5	21	37	53	69	85	101	117	133	149	165	181	197	213	229	245	
6	6	22	38	54	70	86	102	118	134	150	166	182	198	214	230	246	
7	7	23	39	55	71	87	103	119	135	151	167	183	199	215	231	247	
8	8	24	40	56	72	88	104	120	136	152	168	184	200	216	232	248	←Upper
9	9	25	41	57	73	89	105	121	137	153	169	185	201	217	233	249	Limits

EUD DFO/IOM Conversion Table

IOM EUD CODE GROUPS LOWER-UPPER	DFO EUD CODE CONVERSION EU GROUP	SA LINE BIT VALUES 15 14 13 12	11 10 09 08
0-9	0	0 0 0 0	0 - 9
16-25	1	0 0 0 1	0 - 9
32-41	2	0 0 1 0	0 - 9
48-57	3	0 0 1 1	0 - 9
64-73	4	0 1 0 0	0 - 9
80-98	5	0 1 0 1	0 - 9
96-105	6	0 1 1 0	0 - 9
112-121	7	0 1 1 1	0 - 9
128-137	8	1 0 0 0	0 - 9
144-153	9	1 0 0 1	0 - 9
160-169	10	1 0 1 0	0 - 9
176-185	11	1 0 1 1	0 - 9
192-201	12	1 1 0 0	0 - 9
208-217	13	1 1 0 1	0 - 9
224-233	14	1 1 1 0	0 - 9
240-249	15	1 1 1 1	0 - 9
		8 4 2 1	

EUD Code Options

QR20	QR21	ADDRESSING
1	1	2ND 10 DIRECT EU'S
1	0	1ST 10 DIRECT EU'S
0	1	2ND 10 INDIRECT EU'S
0	0	1ST 10 INDIRECT EU'S

QR16	QR17	QR18	QR19	ADDRESSING
0	0	0	0	EU-0
1	0	0	0	EU-1
0	1	0	0	EU-2
1	1	0	0	EU-3
0	0	1	0	EU-4
1	0	1	0	EU-5
0	1	1	0	EU-6
1	1	1	0	EU-7
0	0	0	1	EU-8
1	0	0	1	EU-9
0	1	0	1	EUD ERROR
1	1	0	1	EUD
0	0	1	1	EUD
1	0	1	1	EUD
0	1	1	1	EUD
1	1	1	1	EUD

Disk Address Limitations

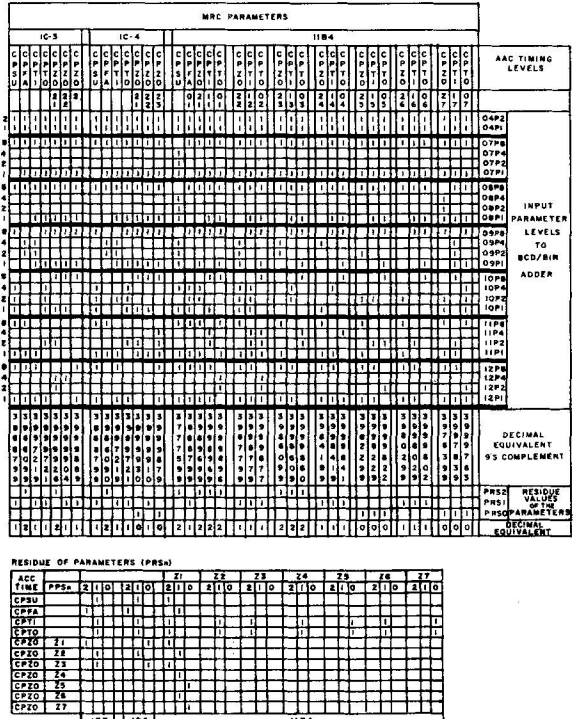
Continuous Segment Numbers / SU No.	SU 0	SU 1	SU 2	SU 3	SU 4	Disk Type
FROM	000,000	111,200	222,400	333,600	444,800	IC3,4
TO	111,199	222,399	333,599	444,599	555,999	
FROM	000,000	222,400	444,800	667,200	889,600	IIB 2,4
TO	222,399	444,799	667,199	889,599	1,111,799	
FROM	000,000	166,800	333,600	500,400	667,200	IIB 6
TO	166,799	333,599	500,399	667,199	833,999	

Multi-Radix Conversion Parameters

PARAMETER	IC3	IC4	IIB 2,4	IIB 6	TALLY REGISTER
SEGM/SU	111200	111200	224000	166800	SU0n
SEGM/FA	13900	13900	55600	41700	FA0n
SEGM/T1	2780	2780	11120	8340	T10n
SEGM/T0	278	278	1112	834	T0nn
SEGM/Z1	73	69	103	103	ZN0n
SEGM/Z2	95	89	122	121	ZN0n
SEGM/Z3	110	120	139	122	ZN0n
SEGM/Z4			158	122	ZN0n
SEGM/Z5			177	122	ZN0n
SEGM/Z6			197	122	ZN0n
SEGM/Z7			216	122	ZN0n

DISK FILE OPTIMIZER

Multi-Radix Conversion Parameter Generation



RESIDUE OF PARAMETERS (PRSn)

ACC TIME	PPSn	Z1			Z2			Z3			Z4			Z5			Z6			Z7		
		2	1	0	2	1	0	2	1	0	2	1	0	2	1	0	2	1	0	2	1	0
CPSU																						
CPFA																						
CPTI																						
CPTO																						
CPZO	Z1																					
CPZO	Z2																					
CPZO	Z3																					
CPZO	Z4																					
CPZO	Z5																					
CPZO	Z6																					
CPZO	Z7																					

IC3 IC4 1184

PRS0 = (Z1+Z3)*IC4+(Z6+Z7)*(TI+TO)

PRS1 = IC3*SU+TI+TO+Z2+Z3)
+IC4*(SU+TI+TO+Z2)
+1184*(Z1*(PA+Z2+Z4+Z6)
+Z2*(TI+TO)
+Z4*(TI+TO)
+Z6*(TI+TO)

PRS2 = IC3*(PA+ZG)+
IC4*(PA!+
(1184*(Z1*(SU+TI+TO+Z1+Z3)
+Z3*(TI+TO)

EU Interface

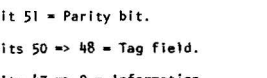
SOURCE	INTERNAL LEVELS	EXTERNAL LEVELS: INDIRECT EU'S, QR21/, QR20/ EU 0-9, ENN3	EXTERNAL LEVELS: INDIRECT EU'S, QR21/, QR20 EU 10-19, ENN4	EXTERNAL LEVELS: DIRECT EU'S, QR21, QR20/ EU 0 9, END/	EXTERNAL LEVELS: DIRECT EU'S, QR21, QR20 EU 10-19, END2
QR16	EU01	EUS16	EUS17	EUS14, 6	EUS15, 7
QR17	EU02	EUS26	EUS27	EUS24, 6	EUS25, 7
QR18	EU03	EUS36	EUS37	EUS34, 6	EUS35, 7
QR19	EU04	EUS46	EUS47	EUS44, 6	EUS45, 7
QR13	SU01	SUS16	SUS17	SUS14, 6	SUS15, 7
QR14	SU02	SUS26	SUS27	SUS24, 6	SUS25, 7
QR15	SU03	SUS36	SUS37	SUS34, 6	SUS35, 7
	ST01	STO16	STO17	STO14, 6	STO15, 7
	ST02	STO26	STO27	STO24, 6	STO25, 7
	ST03	STO36	STO37	STO34, 6	STO35, 7
	ST04	STO46	STO47	STO44, 6	STO45, 7
	ST05	STO56	STO57	STO54, 6	STO55, 7
EU	ST06	STO66	STO67	STO64, 6	STO65, 7
INPUT	ST07	STO76	STO77	STO74, 6	STO75, 7
	ST08	STO86	STO87	STO84, 6	STO85, 7
	ST09	STO96	STO97	STO94, 6	STO95, 7
	ST10	STO106	STO107	STO104, 6	STO105, 7
	ST11	STO116	STO117	STO114, 6	STO115, 7
	ST12	STO126	STO127	STO124, 6	STO125, 7
	EBY	EUBS6	EUBS7	EUBS4, 6	EUBS5, 7
	STRB	STRB6	STRB7	STRB4, 6	STRB5, 7
	SNA	SURY6	SURY7	SURY4, 6	SURY5, 7
CRL		ACGR+ACGR6	ACRQ+ACRQ7	ACGR+ACGR6	ACRQ+ACRQ7
QR00	CTSL	CTSL6	CTSL7	CTSL4, 6	CTSL5, 7
CRL	SEL1			SEL14, 16	
CRL	SEL2				SEL25, 27
CRL	SEL3	SEL36			
CRL	SEL4		SEL47		
LOCATION		JA4	HA4	HB2	HA8
LOCATION		JA5	HA5	JB2	JA8

CRL = CONFLICT RESOLUTION LOGIC

DATA COMMUNICATIONS PROCESSOR

Word Structure

BASIC WORD STRUCTURE

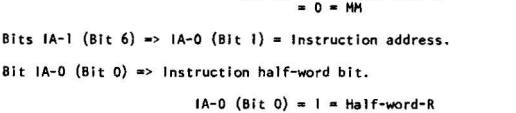


Bit 51 = Parity bit.

Bits 50 => 48 = Tag field.

Bits 47 => 0 = Information.

INSTRUCTION WORD ADDRESS STRUCTURE



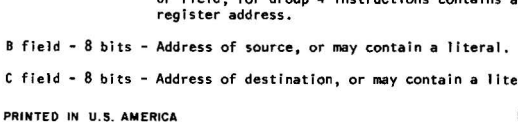
Bits IA-1 (Bit 7) = Instruction source bit.
IA-1 (Bit 7) = 1 = Local Memory
= 0 = MM

Bits IA-1 (Bit 6) => IA-0 (Bit 1) = Instruction address.

Bit IA-0 (Bit 0) => Instruction half-word bit.

IA-0 (Bit 0) = 1 = Half-word-R
= 0 = Half-word-L

INSTRUCTION HALF-WORD FORMAT



Op field - 5 bits - Basic instruction code.

A field - 3 bits - For all but Group 4 instructions, an extension of the OP field; for Group 4 instructions contains a register address.

B field - 8 bits - Address of source, or may contain a literal.

C field - 8 bits - Address of destination, or may contain a literal.

OP Field Codes

OP CODE		A FIELD							
47:1	46:4	0	1	2	3	4	5	6	7
0	0	IDLE							
0	1	BREAK	POINT	POINT	POINT	POINT	POINT	POINT	POINT
0	2	AWI	AWI	AWI	AWI	AWRR	AWRR	AWRR	AWRR
0	3	MSKW	MSKR			SHIFT	HEYU		
0	4	ARWN	ARIN	DBYZ	DBYN	BRAN			
0	5	GOTO [NO]	GOTO [GTR]	GOTO [LSS]	GOTO [NEQ]	GOTO [EQL]	GOTO [GEQ]	GOTO [LEQ]	GOTO [UNC]
0	6	GOX [NO]	GOX [GTR]	GOX [LSS]	GOX [NEQ]	GOX [EQL]	GOX [GEQ]	GOX [LEQ]	GOX [UNC]
0	7	GOI [NO]	GOI [GTR]	GOI [LSS]	GOI [NEQ]	GOI [EQL]	GOI [GEQ]	GOI [LEQ]	GOI [UNC]
0	8	MOVE	LMRI	SMRD	LMRD		LMWI	SMWD	LMWD
0	9		MMR			MMWU	MMWR	MMWP	MMRP
0	C	HAD				HSB			
0	D	HADB				HSBB			
0	E	TRAN [US←E]	TRAN [BCL←E]			TRAN [E←US]	TRAN [E←BCL]		
0	F	PARY [EVEN]	PARY [ODD]						
1	0	SUB							
1	1	LOR							
1	2	LORB							
1	3	LORC							
1	4	LAOM							
1	5	LAN							
1	6	LANB							
1	7	LANC							
1	8	SUBB							
1	9	LEO							
1	A	LEOB							
1	B	LEOC							
1	C	SUBC							
1	D	ADD							
1	E	ADDB							
1	F	ADDC							

A Field is used for addressing on these operators.

NOTE: Blank positions on chart are unused codes. Results of their usage cannot be predicted.

DATA COMMUNICATIONS PROCESSOR

Addressing

A Field

For instructions in groups 0, 1, 2, and 3 the A field is an extension of the OP field. For instructions in group 4 the A field contains a register address as shown below.

A field			Instruction is not "literal in C field" Register selected as source	Instruction is "literal in C field" Register selected as destination
A2	A1	A0		
0	0	0	None (data = zeros)	None (no data stored)
0	0	1	AA	Indirect (The B field selects both the source and the destination)
0	1	0	X	X
0	1	1	Y	Y
1	0	0	D	D
1	0	1	MA0	MA0
1	1	0	MA1	MA1
1	1	1	MA2	MA2

B Field

The B field can contain any of the following items:

Literal

B:C Branch address (groups 0 and 1)

B:C local memory address

register address

Scratchpad memory address (groups 2, 3, and 4)

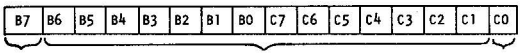
Word register byte address

Indirect address

Literal

The B field contains a literal in most group 0 instructions and in the group 3 and 4 instructions that specify a literal in the B field.

B:C Branch Address Format



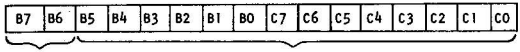
Bit B7 = 0 - Branch to MM
= 1 - Branch to LM

B6 => C1 - Address

C0 = 0 - Branch to half-word L

= 1 - Branch to half-word R

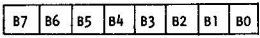
B:C Local Memory Address Format



Bit B7 = 6 = 0

B5 => C0 - Local memory data address

Register Address Format



Bits 7 => 6 = 00 - Register Address ID

5 => 3 = Column address

2 => 0 = Row selection

DATA COMMUNICATIONS PROCESSOR

Register Address Format (Cont)

B [5:3]	B2 = 1	B1 = 1	B0 = 1
000	Zero	Zero	Zero
001	AA	AC	AI
010	D	Y	X
011	MA-2	MA-1	MA-0
100	CF	IA-1	IA-0
101	W-0	W-1	W-2
110	W-3	W-4	W-5
111	reserved	reserved	reserved

Scratchpad Memory Address Format

B7	B6	B5	B4	B3	B2	B1	B0

Bits B7 = B6 = 10 - Scratchpad Memory ID

B5 = B3 = Scratchpad word address

B2 = B0 = Half-word selection bit, or byte designation

Scratchpad Memory Address Format (Cont)

Scratchpad Word Selection

Code			Scratchpad
B5	B4	B3	Memory Word
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Half-word and Byte Selection

Code			Group 2 Half-word Selected	Group 3 Half-word Selected	Group 3 or Group 4 Byte Selected
B2	B1	B0			
0	0	0	L	L	None
0	0	1	L	X	0
0	1	0	L	X	1
0	1	1	L	X	2
1	0	0	R	R	3
1	0	1	R	X	4
1	1	0	R	X	5
1	1	1	R	X	6 (Tag field)

X = Improper Code

DATA COMMUNICATIONS PROCESSOR

Word Register Byte Address Format

B7	B6	B5	B4	B3	B2	B1	B0

Bits B7 => B6 = 11 - Word register byte address ID.

B5 => B3 = 111

B2 => B0 = Single byte designation

Code			Word Register
B2	B1	B0	Byte Selected
0	0	0	None
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Indirect Address Format

B7	B6	B5	B4	B3	B2	B1	B0

Bits B7 => B6 = 01 - Indirect Address ID

B5 => B0 = All zeros

The contents of the Y register are used as the B address.

C Field

The C field can contain the same items as the B field.

Literal

The C field contains a literal in most group 0 instructions and in the group 4 instructions that specify a literal in the C field.

C Address

Contains the address of a destination in group 2 instructions except the local memory direct address instructions, group 3 instructions, and group 4 instructions that do not specify a literal in the C field. If the C field does not select a location, information is not stored.

Register Address Format

Bits C7 = C6 = 00.

Register column and row selection in the same as for B field.

Scratchpad Memory Address Format

(Same as B field).

Word Register Byte Address Format

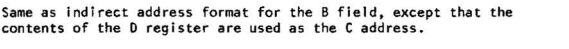
(Same as B field (cont))

Indirect Address Format

Same as indirect address format for the B field, except that the contents of the D register are used as the C address.

INSTRUCTION REGISTER FAULTS

Format

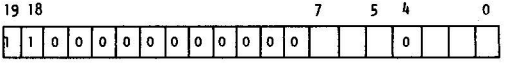


DATA COMMUNICATIONS PROCESSOR

COMMANDS

ADDRESS					FAULT
28	27	26	25	24	
0	0	0	0	0	INITIALIZE FROM SCAN-OUT
0	0	0	0	1	LM DATA ADDRESS INVALID
0	0	0	1	0	LM DATA WORD (W) PARITY ERROR
0	0	0	1	1	ADAPTER CLUSTER ERROR
0	0	1	0	0	MM PROTECTED WRITE DENIED
0	0	1	0	1	MM DATA ADDRESS INVALID
0	0	1	1	0	MM DATA WORD (W) PARITY ERROR
0	0	1	1	1	MM MEMORY-DETECTED ERROR ON DATA ACCESS
0	1	0	0	0	LM INST. TAG ERROR, INVALID OP.
0	1	0	0	1	LM INST. ADDRESS INVALID
0	1	0	1	0	LM INST. (P) PARITY ERROR
0	1	0	1	1	UNEXPECTED TIME-OUT
0	1	1	0	0	MM INST. TAG ERROR, INVALID OP.
0	1	1	0	1	MM INSTRUCTION ADDRESS INVALID
0	1	1	1	0	MM INST. (P) PARITY ERROR
0	1	1	1	1	MM MEMORY DETECTED ERROR ON FETCH
1	0	0	0	0	FAULT FOLLOWING A FAULT: OUT OF CONTROL

ADDRESS/FUNCTION CODE FORMAT



Bits 7 => 5 = Function code

000 = Initialize

010 = Halt

100 = Set Attention Needed

3 => 1 = DCP Address

0 = Address significance

0 = DCP address not significant; all DCP's to respond

1 = Only the addressed DCP to respond

SCAN OUT INFORMATION WORD

47	43	39	35	31	27	23	19	15	11	7	3
46	42	38	34	30	26	22	18	14	10	6	2
45	41	37	33	29	25	21	17	13	9	5	1
44	40	36	32	28	24	20	16	12	8	4	0

DATA COMMUNICATIONS PROCESSOR

BRANCH RELATIVE CONDITIONS

TYPE OF BRANCH	RESULT INDICATORS		
	CF0		CF1
NO (1)	NA		NA
GTR	0	and	0
LSS	1		NA
NEQ	NA		0
EQL	NA		1
GEQ	0		NA
LEQ	1	or	1
UNC (2)	NA		NA

NA - Not applicable, may be 0 or
(1) - Never Branches (No Op)
(2) - Always Branches

SHIFT MA OPTIONS

B AND C FIELD	OPTION
B [7:2] ≠ 01	B [2:3] = Number of Shifts
B [7:2] = 01	Y Reg. = Number of Shifts
C [5:1] = 1	Byte Circulate
C [4:1] = 1	End around shift
C [3:1] = 1	Concatenate MA
C [2:3]	MA Register row select

MA Concatenations

Bit Configuration						
C5	C4	C3	C2	C1	C0	Shift
0	0	1	0	1	1	MA1:MA0 (":" denotes MA1 [0:1] => MA0 [7:1])
0	0	1	1	0	1	MA2:MA0 (MA1 bypassed)
0	0	1	1	1	0	MA2:MA1
0	0	1	1	1	1	MA2:MA1:MA0
0	1	1	0	1	1	MA1:MA0; MA0 [0:1] => MA1 [7:1]
0	1	1	1	0	1	MA2:MA0; MA0 [0:1] => MA2 [7:1]
0	1	1	1	1	0	MA2:MA1; MA1 [0:1] => MA2 [7:1]
0	1	1	1	1	1	MA2:MA1:MA0; MA0 [0:1] => MA2 [7:1]
0	1	0	0	1	1	MA1, MA0; MA0 [0:1] => MA1 [7:1]
0	1	0	1	0	1	MA2, MA0; MA0 [0:1] => MA2 [7:1]
0	1	0	1	1	0	MA2, MA1; MA1 [0:1] => MA2 [7:1]
0	1	0	1	1	1	MA2, MA1, MA0; MA0 [0:1] => MA2 [7:1]

Translate Operator A Field Codes

A2	A1	A0	Translation
0	0	0	EBCDIC to USASCII
0	0	1	EBCDIC to BCL (Burroughs Common Language)
0	1	0	(reserved)
0	1	1	(reserved)
1	0	0	USASCII to EBCDIC
1	0	1	BCL to EBCDIC
1	1	0	(reserved)
1	1	1	(reserved)

DATA COMMUNICATIONS PROCESSOR

Adapter Clusters

INTERRUPTS

Types

```
CC [4:5] = 00001 - DATA PRESENT
CC [4:5] = 01001 - BYTE REQUEST
CC [4:5] = 01011 - PROGRAM TIMER
CC [4:5] = 00010 - CONTROL
```

Formats

```
DATA PRESENT    876543210  HD  DES
(DC = CHARACTER)   00001   CC
                XXXXXXXXX  DC
-------------------------------
BYTE REQUEST    876543210  HD  DES
                   01001   CC
                000000000  DC
-------------------------------
PROGRAM TIMER   876543210  HD  DES
(DC = INFO-NOT     01011   CC
SIGNIFICANT)    00XXXXXXX  DC
-------------------------------
CONTROL         876543210  HD  DES
(C = COMMAND       00010   CC
I = INTERRUPT)  00CCC:III  DC
```

Voice Response - Type 5

	87	654	3210	DC
	00	001	1111	PROGRAM ERROR
	00	011	1010	PROGRAM ERROR
	00	011	1011	PROGRAM ERROR
┌→	00	101	1001	BUFFER EMPTIED
TRANSMIT	00	101	1010	BYTE REQUEST BACK-UP
	00	101	1011	3 SEC NOT READY
	00	101	1111	RDY OFF
→T	00	111	1111	PROGRAM ERROR

Asynchronous - Type 1 or 2

	87	654	3210	DC
RING	00	001	1111	RING INDICATOR
R→	00	011	0110	RECEIVE STATUS
E	00	011	0111	INTER CHAR TIME OUT
C	00	011	1000	STOP BIT ERROR
E	00	011	1001	BUFFER OVERFLOW
I	00	011	1010	30 SEC TIME OUT
V	00	011	1011	3 SEC NOT READY
E	00	011	1100	BREAK
	00	011	1101	LONG SPACE DISC
	00	011	1110	CF OFF
└→	00	011	1111	CC OFF
T→	00	101	1010	BYTE REQUEST BACK-UP
R				
A	00	101	1011	3 SEC NOT READY
N				
S	00	101	1100	BREAK
M				
I	00	101	1101	LONG SPACE DISC
T				
	00	101	1110	CB OFF
└→	00	101	1111	CC OFF
TRNS BREAK	00	111	1111	TRANSMITTING BREAK

DATA COMMUNICATIONS PROCESSOR

Auto Call - Type 3

	87	654	3210	DC
	00	001	1111	PROGRAM ERROR
	00	011	1011	PROGRAM ERROR
	00	101	1010	BYTE REQUEST BACK-UP
	00	101	1011	3 SEC NOT READY
	00	101	1100	ACR ON
	00	101	1101	OLD OR COS OFF
	00	101	1110	COS ON
	00	101	1111	PWR OFF
	00	111	1111	PROGRAM ERROR

Synchronous - Type 1 or 2

		87	654	3210	DC
RING		00	001	1111	RING INDICATOR
	→	00	011	0110	RECEIVE STATUS
RECEIVE		00	011	0111	3 SEC NO SYNCS
		00	011	1001	BUFFER OVERFLOW
		00	011	1010	30 SEC TIMEOUT
		00	011	1011	3 SEC NOT READY
		00	011	1110	CF OFF
	→	00	011	1111	CC OFF
	→	00	101	1001	SYNC FILL
		00	101	1010	BYTE REQUEST BACK-UP
TRANSMIT		00	101	1011	3 SEC NOT READY
		00	101	1110	CB OFF
	→	00	101	1111	CC OFF
TRNS BREAK		00	111	1111	TRANSMITTING BREAK

Touchtone - Type 4

	87	654	3210	DC
	00	001	1111	RING INDICATOR
	→ 00	011	0110	RECEIVE STATUS
RECEIVE	00	011	1001	BUFFER OVERFLOW
	00	011	1010	30 SEC TIMEOUT
	00	011	1011	3 SEC NOT READY
	→ 00	011	1111	CC OFF
	→ 00	101	1001	VOICE RES ENABLE
TRANSMIT	00	101	1010	BYTE REQUEST BACK-UP
	00	101	1011	3 SEC NOT READY
	00	101	1111	CC OFF
	→ 00	111	1111	PROGRAM ERROR

Interrupt Codes and Formats

Interrupt	AI [8:9]	AC [4:5]
Data Present	C2 [8:9]	00001
BC/BI	00 BC [2:2] 1 BI [3:4]	00010
Byte Request	0 0 0 0 0 0 0 0 0 0	01001
PT	0 0 0 0 0 0 0 0 0 0	01011

DATA COMMUNICATIONS PROCESSOR

COMMANDS

Byte Address

Byte	Field	Length(A1)	Miscellaneous
0	Type	[4:5]	[5:1] = F I P
1	C2	[8:9]	CC3F = C2QF
2	BC/B1	[6:3] [3:4]	CC3F = CLEAN HOUSE
3	SC/SA	[4:2] [2:3]	COOF=[5:1]C1OF=[6:1]
4	CT/BT	[8:2] [6:7]	
5	C1	[8:9]	
6	IR	[8:9]	CC3F = IR9F
7	MR	[6:7]	
3MAC3F	PT	[7:8]	WRITE PT ONLY
7MAC3F	QR	[7:2][4:2][1:2]	INT OR ONLY

Type Field

A1 (4:5)	DEVICE
00010	TOUCH TONE
00100	VOICE RESPONSE_TRIPLE LIGHT
00101	VOICE RESPONSE_SINGLE LIGHT
00111	AUTOMATIC CALLING UNIT
01BBB	SYNCHRONOUS
1CCCC	ASYNCHRONOUS

Command Field

AC [4:5]	A1 [6:3]	COMMAND
01010	000	OUT OF SERVICE
01010	001	LOOK FOR CE
01010	010	RECEIVE READY
01010	011	INITIATE RECEIVE
01010	100	TRANSMIT READY
01010	101	INITIATE TRANSMIT
01010	111	BREAK
00010	110	FINISH TRANSMIT
00010	000	IDLE

Program Timer Format

DC FIELD = ORRRCCCCC		RATE
RANGE - MSEC	RATE PERIOD	CODE
0064 - 192	0.0064	000
192 - 1536	0.0312	001
1536 - 12 288	0.4096	010
12 288- 981	3.27	011
981 - 786	26.2	100
786 - 6291	209.7	101
6291 - 50310	1677.	110
50310 - 402630	13421	111

Synchronous Type Field

01BBB	A10F - A18F
0100-	011010000
0101-	010011000
0110-	0101000
0111-	1010110

TYOF = 0 RECEIVE - EDIT ALL SYNCS
TRANSMIT- INHIBIT SYNC
FILL INTERRUPT
TYOF = 1 RECEIVE - EDIT ONLY 1ST.
TWO SYNCS
TRANSMIT- ENABLE SYNC
FILL INTERRUPT

Asynchronous Type Fields

A1 [4:5]	BAUD RATE	CHAR SIZE
10001	45.5	7.5
10010	56.9	7.5
10011	75	7.5
10100	110	11
10101	134.5	9
10110	150	10
10111	300	10
11000	600	10
11001	1200	10
11010	1200	6
11011	1800	10
11100	2400	10
11101	3600	10
11110	4900	10
11111	9600	10

Interrogate Codes and Formats

Interrogate	AI [8:9]	AC [4:5]
TY	000 FIP TY [4:5]	10000
C2	C2 [8:9]	1a001
BC/BI	00 BC [2:3] BI [3:4]	10010
SC/SA	00 C110 C100 SC [1:2] SA [2:3]	10011
CT/BT	CT [1:2] BT [6:7]	10100
C1	C1 [9:9]	10101
IR	IR [8:9]	1b110
MR	00 MR [6:7]	10111
C2 clear**	C2 [8:9]	11001
BC/BI control*	00 BC [2:2] 1 BI [3:4]	11010
OR	000 OR [5:6]	11111

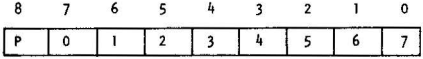
a = C2 occupied (zero before interrogate)

b = IR9 (zero before interrogate)

* This interrogate is identical to the BC/BI interrogate except that BC0 is set to one forcing a BC/BI interrupt condition for the software.

Data Formats

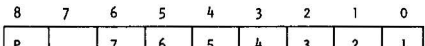
EBCDIC CODED DATA (CHARACTER)



Bits 8 = Parity bit.

7 => 0 [7:8] = Data bits.

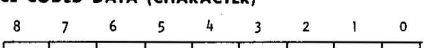
USASCII CODED DATA (CHARACTER)



Bits 8 = Parity bit.

6 => 0 = Data bits.

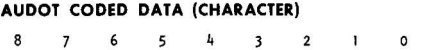
BCL CODED DATA (CHARACTER)



Bits 8 = Parity bit.

5 => 0 = Data bits.

BAUDOT CODED DATA (CHARACTER)



Bits 5 = Shift bit.

4 => 0 = Data bits.

DATA COMMUNICATIONS PROCESSOR

PTT CODE/6 CODED DATA (CHARACTER)

8	7	6	5	4	3	2	1	0	Bit designation.
C			B	A	8	4	2	1	

Bits 8 = Parity bit.

5 => 0 = Data bits.

XS-3 CODED DATA (CHARACTER)

8	7	6	5	4	3	2	1	0	Bit designation.
P			6	5	4	3	2	1	

Bits 8 = Parity bit.

5 => 0 = Data bits.

BURROUGHS NUMERIC CODED DATA (CHARACTER)

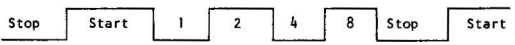
8	7	6	5	4	3	2	1	0	Bit designation.
					8	4	2	1	

Bits 3 => 0 = Data bits.

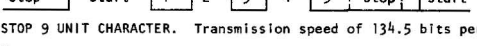
Communication Line Character Formats

SERIAL-BY-BIT CHARACTER FORMATS

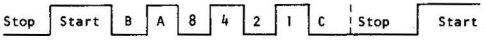
START-STOP 6 UNIT CHARACTER. Transmission speed is 1200 bits per second.



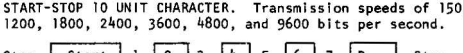
START-STOP 7.5 UNIT CHARACTER. Transmission speeds of 45.5, 56.9 and 75 bits per second.



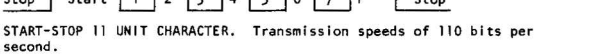
START-STOP 9 UNIT CHARACTER. Transmission speed of 134.5 bits per second.



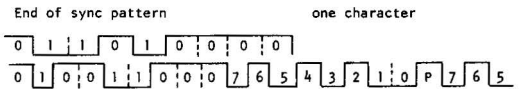
START-STOP 10 UNIT CHARACTER. Transmission speeds of 150, 300, 600, 1200, 1800, 2400, 3600, 4800, and 9600 bits per second.



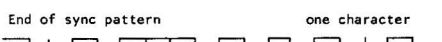
START-STOP 11 UNIT CHARACTER. Transmission speeds of 110 bits per second.



SYNC PATTERN 9 BIT CHARACTER. Transmission speeds of 2000, 2400, 4800, and 9600 bits per second.

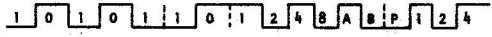


SYNC PATTERN EIGHT-BIT CHARACTERS. Transmission speeds of 2000, 2400, 4800, and 9600 bits per second.



DATA COMMUNICATIONS PROCESSOR

SYNC PATTERN SEVEN-BIT CHARACTER. Transmission speeds of 2000, 2400, 4800, and 9600 bits per second.



PARALLEL-BY-BIT CHARACTER FORMATS.

DIALING DIGIT BITS (OUTPUT). Transmission speed is controlled by the Automatic Calling Unit.

Number Bit 1 ----

Number Bit 2 ----

Number Bit 4 ----

Number Bit 8 ----

TOUCH TONE CHARACTER 4 BITS (INPUT). Transmission speed is a maximum of 14 input characters per second.

----- Bit 1

----- Bit 2

----- Bit 3

----- Bit 4

Comparison Register Flip-Flop Designations

Bit Position	Flip-Flop Designation	Special Function
0	116 (CF0)	Carry flip-flop. Controlled by the carry/borrow signal from the arithmetic unit (when it is used). Cleared when a branch occurs or by a Shift MA.
1	117 (CF1)	Zero-result flip-flop. Controlled by the result = zero signal from the arithmetic unit (when it is used). Cleared when a branch occurs or by a Shift MA.
2	118	Internal Carry flip-flop. Used in place of 116 (CF0) during the IA + IBA addition preceding a MM fetch and also during a Halfword Add or Subtract. Otherwise zero.
3	119	Fault Control flip-flop. Set when the special fault instruction is executed. If 119 = 1, the fault interrupt action is modified.
4	120	IA Counted flip-flop. Set when the address is counted up normally; otherwise cleared when the instruction is completed.
5	121	AI parity bit. Controlled by the A18 signal from the cluster interface and by the Parity instruction.
6	122	MM interlock flip-flop. Must be zero. (If a MM access is separately buffered, 122 = 1 defeats the "wait" which occurs when the MA or W register is addressed before the access is completed).
7	123	Main System Attention Needed flip-flop. Set by the scan-out of Set Attention Needed. Cleared when it causes a branch in the instructions which explicitly test it.

DATA COMMUNICATIONS PROCESSOR

COMPARE BITS (CF1 AND CF0) CONDITIONAL RESULTS

CF1	CF0	Compare Condition	Other Significance
0	0	>	Cleared state
0	1	<	Carry or borrow
1	0	=	Zero result
1	1		Carry and zero result (unlikely)

CARRY FLIP-FLOP CHART

	CF0=0	CF0=1	CF1=0	CF1=1
IDLE	N/A	N/A	N/A	N/A
AWI	N/A	N/A	N/A	N/A
AWRR	N/A	N/A	N/A	N/A
BKP	Branch Cleared	Branch Cleared	Branch Cleared	Branch Cleared
SHFT	Cleared	Cleared	Cleared	Cleared
HEYU	N/A	N/A	N/A	N/A
MSKW	N/A	N/A	N/A	N/A
MSKR	N/A	N/A	N/A	N/A
ARWN	Branch Cleared	Branch Cleared	Branch Cleared	Branch Cleared
ARIN	Branch Cleared	Branch Cleared	Branch Cleared	Branch Cleared
DBYZ	Branch Cleared	Branch Cleared	Branch Cleared	Branch Cleared
DBYN	Branch Cleared	Branch Cleared	Branch Cleared	Branch Cleared
BRAN	Branch Cleared	Branch Cleared	Branch Cleared	Branch Cleared
GOTO	Branch Cleared	Branch Cleared	Branch Cleared	Branch Cleared
GOX	Branch Cleared	Branch Cleared	Branch Cleared	Branch Cleared
GOI	Branch Cleared	Branch Cleared	Branch Cleared	Branch Cleared

Carry Flip-Flop Chart (Cont)

	CF0=0	CF0=1	CF1=0	CF1=1
MOVE	N/A	N/A	N/A	N/A
LMRD	N/A	N/A	N/A	N/A
LMRI	N/A	N/A	N/A	N/A
LMWD	N/A	N/A	N/A	N/A
LMWI	N/A	N/A	N/A	N/A
SMRD	N/A	N/A	N/A	N/A
SMWD	N/A	N/A	N/A	N/A
MMR	N/A	N/A	N/A	N/A
MMWU	N/A	N/A	N/A	N/A
MMWR	N/A	N/A	N/A	N/A
MMWP	N/A	N/A	N/A	N/A
MWRP	N/A	N/A	N/A	N/A
HAD	No Carry	Carry	Nonzero Result	Zero Result
HADB	No Carry	Carry	Nonzero Result	Zero Result
HSB	No Carry	Carry	Nonzero Result	Zero Result
HSBB	No Carry	Carry	Nonzero Result	Zero Result
TRAN	N/A	N/A	N/A	N/A
PARY	Cleared	Cleared	Parity Error	Good Parity
ADD	No Carry	Carry	Nonzero Result	Zero Result
ADDB	No Carry	Carry	Nonzero Result	Zero Result
ADDC	No Carry	Carry	Nonzero Result	Zero Result
SUB	No Borrow	Borrow	Nonzero Result	Zero Result
SUBB	No Borrow	Borrow	Nonzero Result	Zero Result
SUBC	No Borrow	Borrow	Nonzero Result	Zero Result
LAN	Cleared	Cleared	Nonzero Result	Zero Result
LANB	Cleared	Cleared	Nonzero Result	Zero Result

DATA COMMUNICATIONS PROCESSOR

Carry Flip-Flop Chart (Cont)

	CF0=0	CF0=1	CF1=0	CF1=1
LANC	Cleared	Cleared	Nonzero Result	Zero Result
LAOM	Cleared	Cleared	Nonzero Result	Zero Result
LOR	Cleared	Cleared	Nonzero Result	Zero Result
LORB	Cleared	Cleared	Nonzero Result	Zero Result
LORC	Cleared	Cleared	Nonzero Result	Zero Result
LEO	Cleared	Cleared	Nonzero Result	Zero Result
LEOB	Cleared	Cleared	Nonzero Result	Zero Result
LEOC	Cleared	Cleared	Nonzero Result	Zero Result

SECTION 8

MAINTENANCE INFORMATION

SECTION 8 — CONTENTS

The terms "Multiplexor" and "I/O Processor" are synonymous.

SECTION 8

MAINTENANCE INFORMATION

SYSTEM

ON-LINE MAINTENANCE AND TEST (MAT) LANGUAGE

The B 6700 on-line maintenance system was developed to enable maintenance and test (CONFIDENCE/DIAGNOSTIC) routines to be performed on the units within the B 6700 system while the system remained on line. Two facilities are incorporated within the on-line maintenance module of the B 6700 Master Control Program to accomplish this objective:

a. a Maintenance And Test language "compiler," which allows the field engineer to exercise the unit(s) in order to perform various adjustments, alignments, and trouble analysis tests, and

b. a set of fixed, built-in test routines (confidence tests) which establish that the disk file and magnetic tape units are operational and functioning properly.

The purpose of the Maintenance And Test (MAT) language is to provide a field engineer with an easy method of communicating with the on-line maintenance module, a tool embedded in the B 6700 MCP. The MAT language is a programming language similar to other source languages in that it is introduced into the computer as data for a "compiler" which subsequently produces executable program object code. The compiler checks the syntax of the input statements and produces a pseudo-machine-language code "program" if no syntax errors are detected. This program is then "executed" by the computer.

The present discussion provides an example of an input card deck bearing MAT language instructions, an example of MAT instruction input from the input display device, a reference listing of the valid syntax for instructions written in this language, and an index of MAT metalinguistic variables.

SAMPLE CARD INPUT DECK

```
<!> END                                                    (C)
VERIFY DISK DK32 (SEGMENT 20000 FOR 5, SELECT ALL)         (B)
<!> SCR; EBCDIC MATLINGO                                   (A)
```

(A) is an MCP control card which calls the system confidence routines and MAT compiler and specifies that the following card(s) carry MAT instructions punched in EBCDIC format. <!> represents an invalid character punched in column 1.

(B) is an example of a card bearing a MAT instruction.

(C) indicates an end of the input card deck.

SYSTEM

INPUT FROM INPUT DISPLAY UNIT

When the input display unit is used as the initiation device, the user must begin the input with "SCR", followed by the MAT source program declarations and/or statements. The input must be ended with the ETX ("End of Text") character.

For example:

```
SCR STATUS MT2 (ETX)
```

SYNTAX FOR MAT LANGUAGE

Two listings are provided in logical order of the complete syntax for the MAT language. The first listing is an index of MAT metalinguistic variables arranged in alphabetical order with references to the line number where each variable is defined in the syntax for the language which is the second listing.

INDEX OF MAT METALINGUISTIC VARIABLES

The following is a listing in alphabetic order of the metalinguistic variables defined in the following list of syntactic definitions. The number to the left of each variable in the present list is the line number of the definition of that variable in the following list.

SYSTEM

LINE NUMBER	ITEM	LINE NUMBER	ITEM
79	<increment>	113	<read SC>
108	<IOlength modifier>	169	<read SC modifier>
133	<IOlength units>	138	<read SC modifier list>
19	<iteration statement>	87	<read statement>
58	<key statement>	50	<relation>
155	<leader modifier>	70	<relational operator>
35	<letter>	160	<relative disk address>
98	<logical operator>	106	<repeat count>
3	<MAT program>	164	<repeat part>
149	<memory protect modifier>	151	<result modifier>
8	<number>	88	<rewind statement>
194	<octal digit>	78	<ripple direction>
193	<octal string>	59	<ripple statement>
66	<operational results>	60	<rotate statement>
5	<output device>	1	<SCR message>
2	<output part>	179	<segment address>
115	<paper motion>	178	<segment type>
156	<parity modifier>	157	<segment type/address modifier>
86	<path statement>		
72	<pattern>	82	<separator>
129	<pattern element>	46	<set statement>
54	<pattern identifier>	61	<shift statement>
103	<pattern list>	131	<single space>
150	<pattern modifier>	89	<skip statement>
53	<pattern part>	105	<space>
71	<pattern size>	170	<space amount>
38	<pattern source>	90	<space statement>
52	<primary>	140	<spacing modifier>
28	<range specification>	158	<stacker modifier>
109	<read CR>	122	<standard I/O modifier>
134	<read CR modifier list>	10	<statement>
110	<read DK>	7	<statements>
166	<read DK modifier>	91	<status statement>
135	<read DK modifier list>	196	<string>
111	<read MT>	25	<subroutine body>
167	<read MT modifier>	14	<subroutine declaration>
136	<read MT modifier list>	24	<subroutine identifier>
112	<read PR>	152	<tag modifier>
168	<read PR modifier>	177	<tag part>
137	<read PR modifier list>	162	<tape test>

LINE NUMBER	ITEM
127	<tape test list>
187	<tape test modifier>
183	<tape test modifier list>
102	<tape test parameters>
189	<tape test trace option>
182	<test number>
92	<test statement>
163	<testape modifier>
128	<testape modifier list>
184	<testape trace option>
69	<time units>
153	<translate modifier>
17	<unconditional statement>
123	<unique I/O modifier>
15	<unit declaration>
26	<unit designator>
39	<unit mnemonic>
99	<unit or file specifier>
101	<unit specifier>
16	<variable declaration>
27	<variable identifier>
67	<verify disk statement>
47	<verify statement>
68	<verify tape statement>
114	<wait part>
48	<wait statement>
34	<while statement>
116	<write CP>
171	<write CP modifier>
141	<write CP modifier list>
117	<write DK>
172	<write DK modifier>
142	<write DK modifier list>
118	<write LP>
173	<write LP modifier>
143	<write LP modifier list>
119	<write MT>
174	<write MT modifier>
144	<write MT modifier list>
120	<write PP>
175	<write PP modifier>
145	<write PP modifier list>
121	<write SC>
176	<write SC modifier>
146	<write SC modifier list>
93	<write statement>
94	<write tapemark statement>

MAT LANGUAGE SYNTAX

The following is the complete syntax for the MAT language. A reference line number is provided to the left of each syntactical formula for cross-reference from the alphabetical index of metalinguistic variables and within the listing. The number(s) to the right of each syntactical formula is (are) the reference line number(s) corresponding to the definition(s) of the various items in the formula.

```
1   <SCR message> ::= SCR <output part> <MAT program>                    2  3

2   <output part> ::= <empty> |                                          4
                      OUTPUT ON <output device>;                         5

3   <MAT program> ::= <declarations>;<statements>                        6  7

4   <empty> ::= {the null string of symbols}

5   <output device> ::= LP <number> |                                    8
                        SC <number>                                      8

6   <declarations> ::= <declaration> |                                   9
                       <declarations>;<declaration>                      6  9

7   <statements> ::= <statement> |                                       10
                     <statements>;<statement>                           7  10

8   <number> ::= <digit> |                                               11
                 <number> <digit>                                       8  11

9   <declaration> ::= <empty>|                                           4
                      <buffer declaration> |                            12
                      <file declaration> |                              13
                      <subroutine declaration>|                         14
                      <unit declaration> |                              15
                      <variable declaration> |                          16

10  <statement> ::= <unconditional statement> |                          17
                    <conditional statement> |                           18
                    <iteration statement>                               19

11  <digit> ::= 0|1|2|3|4|5|6|7|8|9

12  <buffer declaration> ::= BUFFER <identifier> = <buffer specs>        20  21
                             <fill part>                                22
```

13	<file declaration> ::= FILE <identifier> = <file name>	20	23
14	<subroutine declaration> ::= SUBROUTINE <subroutine identifier>; <subroutine body>	24 25	
15	<unit declaration> ::= UNIT <identifier> = <unit designator>	20	26
16	<variable declaration> ::= VARIABLE <variable identifier> <range specification>	27 28	
17	<unconditional statement> ::= <basic statement> | <compound statement>	29 30	
18	<conditional statement> ::= <if statement>	31	
19	<iteration statement> ::= <do statement> | <for statement> | <while statement>	32 33 34	
20	<identifier> ::= <letter>| <identifier> <letter>	35 20	35
21	<buffer specs> ::= <buffer size> <frame size>	36	37
22	<fill part> ::= <empty> | , FILL <pattern source>	4 38	
23	<file name> ::= "BADDISK/ . . ."		
24	<subroutine identifier> ::= <identifier>	20	
25	<subroutine body> ::= <statement>	10	
26	<unit designator> ::= <unit mnemonic> <number>	39	8
27	<variable identifier> ::= <identifier>	20	

```
28  <range specification> ::= <empty> |                                  4
                              , RANGE = <allowable range>                40

29  <basic statement> ::= <buffer statement> |                           41
                          <call statement> |                           42
                          <comment statement> |                        43
                          <display statement> |                        44
                          <I/O statement> |                            45
                          <set statement> |                            46
                          <verify statement> |                         47
                          <wait statement> |                           48

30  <compound statement> ::= BEGIN <compound tail>                       49

31  <if statement> ::= IF <relation> THEN <statement> <else part>        50  10  51

32  <do statement> ::= DO <statement> UNTIL <relation>                   10  50

33  <for statement> ::= FOR <primary> DO <statement>                     52  10

34  <while statement> ::= WHILE <relation> DO <statement>                50  10

35  <letter> ::= A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|
                 S|T|U|V|W|X|Y|Z

36  <buffer size> ::= <number>  CHARACTERS |                             8
                      <number>  WORDS                                    8

37  <frame size> ::= <empty> |                                           4
                     IN BCL |
                     IN EBCDIC

38  <pattern source> ::= WITH PATTERN <pattern part> |                   53
                         USING PATTERN <pattern identifier>             54

39  <unit mnemonic> ::= CP|CR|DK|LP|MT|PP|PR|SC
```

```
40  <allowable range> ::= <number> THRU <number> |                    8   8
                          FILE <identifier>                           20

41  <buffer statement> ::= <COMPARE statement>|                       55
                           <DUMP statement>|                        56
                           <FILL statement>|                        57
                           <KEY statement>|                         58
                           <RIPPLE statement>|                      59
                           <ROTATE statement>|                      60
                           <SHIFT statement>                        61

42  <call statement> ::= CALL <subroutine identifier>                 24

43  <comment statement> ::= COMMENT
                            {any sequence of EBCDIC
                            characters excluding semi-
                            colon and ETX}

44  <display statement> ::= DISPLAY ( <display item list> )           62

45  <I/O statement> ::= <basic I/O statement> <I/O modifier list>     63  64
                        <I/O error disposition>                     65

46  <set statement> ::= SET <variable identifier> = <operational results>  27  66

47  <verify statement> ::= <verify disk statement>|                   67
                           <verify tape statement>                  68

48  <wait statement> ::= WAIT <primary> <time units>                  52  69

49  <compound tail> ::= <statement> END |                             10
                        <statement> ; <compound tail>               10  49

50  <relation> ::= <primary> <relational operator> <primary>          52  70  52
```

```
51  <else part> ::= <empty> |                                          4
                    ELSE  statement                                    10

52  <primary> ::= <number>                                             8
                  <variable identifier>|                               27
                   HEX "<hex string>" |                                191
                   OCT "<octal string>"|                               193
                   HIGHRANGEOF ( <variable identifier> )|              27
                   LOWRANGEOF ( <variable identifier> ) |              27
                   CURRENTTIME

53  <pattern part> ::= <pattern identifier> <pattern size> <pattern>   54  71  72

54  <pattern identifier> ::= <empty>|                                  4
                             <identifier>                              20

55  <compare statement> ::= COMPARE BUFFER <identifier> <buffer offset> TO   20  73
                            BUFFER <identifier> <buffer offset>              20  73
                            <area length> <frame size>                       74  37
                            <compare error disposition>                      75

56  <dump statement> ::= DUMP BUFFER <identifier> <buffer part>        20  76
                         <dump data format>                            77

57  <fill statement> ::= FILL BUFFER <identifier> <pattern source>     20  38

58  <key statement> ::= KEY BUFFER <identifier> CHARACTER <primary> FOR   20  52
                        <primary> WITH <primary>                         52  52

59  <ripple statement> ::= ripple buffer <identifier> <buffer part>    20  76
                           <ripple direction> <increment>               78  79

60  <rotate statement> ::= ROTATE BUFFER <identifier> <buffer part>    20  76
                           <direction> <increment>                      80  79
```

```
61  <shift statement> ::= SHIFT BUFFER <identifier> <buffer part>                20  76
                          <direction> <increment>                                80  79

62  <display item list> ::= <display item>|                                       81
                            <display item list> <separator>                     62  82
                            <display item>                                      81

63  <basic I/O statement> ::= <backspace statement>|                              83
                              <check statement>|                                84
                              <erase statement>|                                85
                              <path statement>|                                 86
                              <read statement>|                                 87
                              <rewind statement>|                               88
                              <skip statement>|                                 89
                              <space statement>|                                90
                              <status statement>|                               91
                              <test statement>|                                 92
                              <write statement>|                                93
                              <write tapemark statement>                        94

64  <I/O modifier list> ::= <empty>|                                              4
                            <I/O modifier>|                                     95
                            <I/O modifier list> <separator>|                    64  82
                            <I/O modifier>                                      95

65  <I/O error disposition> ::= <empty>|                                          4
                                ON ERROR ( <I/O error action list> )            96

66  <operational results> ::= <primary>|                                         52
                              <primary> <adding operator> <primary>|            52  97  52
                              <primary> <logical operator> <primary>            52  98  52

67  <verify disk statement> ::= VERIFY DISK <unit or file specifier>              99
                                ( <disk test parameters> )                      100
```

```
68  <verify tape statement> ::= VERIFY TAPE <unit specifier>                    101
                                ( <tape test parameters> )                      102

69  <time units> ::= SECONDS|MILLISECONDS

70  <relational operator> ::= >|<|=

71  <pattern size> ::= <number>                                                 8

72  <pattern> ::= <empty>|                                                      4
                  ( <pattern list> )                                            103

73  <buffer offset> ::= <empty>|                                                4
                        CHARACTER <primary>                                     52

74  <area length> ::= <empty>|                                                  4
                      FOR <primary>                                             52

75  <compare error disposition> ::= <empty>|                                    4
                                    ON ERROR
                                    ( <compare error action list> )             104

76  <buffer part> ::= <buffer offset> <area length>                             73  74

77  <dump data format> ::= IN HEX|IN OCTAL

78  <ripple direction> ::= MINUS|PLUS

79  <increment> ::= <empty>|                                                    4
                    <number>|                                                   8
                    + <number>|                                                 8
                    - <number>                                                  8

80  <direction> ::= LEFT|RIGHT
```


PRINTED IN U.S. AMERICA 5000276

```
81  <display item> ::= <primary>|                                          52
                       " <string> "                                       196

82  <separator> ::= <space>|,                                              105

83  <backspace statement> ::= BACKSPACE <unit specifier> <repeat count>    101 106
                              <I/O error disposition>                     65

84  <check statement> ::= CHECK <unit or file specifier>                   99
                          <check modifier list> <I/O error disposition>   107 65

85  <erase statement> ::= ERASE <unit specifier> <I/O length modifier>     101 108
                          <I/O modifier list> <I/O error disposition>     64  65

86  <path statement> ::= PATH <unit or file specifier>                     99

87  <read statement> ::= <read CR>|                                        109
                         <read DK>|                                       110
                         <read MT>|                                       111
                         <read PR>|                                       112
                         <read SC                                         113

88  <rewind statement> ::= REWIND <unit specifier> <wait part>             101 114

89  <skip statement> ::= SKIP <unit specifier> <paper motion>              101 115
                         <I/O error disposition>                          65

90  <space statement> ::= SPACE <unit specifier> <repeat count>            101 106
                          <I/O error disposition>                         65

91  <status statement> ::= STATUS <unit specifier>                         101

92  <test statement> ::= TEST <unit specifier>                             101
```

```
93  <write statement> ::= <write CP>|                                    116
                          <write DK>|                                    117
                          <write LP>|                                    118
                          <write MT>|                                    119
                          <write PP>|                                    120
                          <write SC>                                     121

94  <write tapemark statement> ::= WRITE <unit specifier> TAPEMARK       101
                                   <I/O error disposition>               65

95  <I/O modifier> ::= <standard I/O modifier>|                          122
                       <unique I/O modifier>                            123

96  <I/O error action list> ::= <I/O error action>|                      124
                                <I/O error action list> <separator>      96  82
                                <I/O error action>                       124

97  <adding operator> ::= +|-

98  <logical operator> ::= AND|OR

99  <unit or file specifier> ::= <unit specifier>|                       101
                                 FILE <identifier>                      20

100 <disk test parameters> ::= <disk address part> <disk test part>      125 126

101 <unit specifier> ::= <unit designator>|                              26
                         UNIT <identifier>                               20

102 <tape test parameters> ::= SELECT <tape test list>|                  127
                               TESTAPE <testape modifier list>           128

103 <pattern list> ::= <pattern element>|                                129
                       <pattern list> <separator> <pattern element>      103 82 129
```

```
104 <compare error action list> ::= <compare error action>|                    130
                                    <compare error action list>                104
                                    <separator> <compare error action>         82  130

105 <space> ::= <single space>|                                                131
                <space> <single space>                                         105 131

106 <repeat count> ::= <primary> RECORDS                                       52

107 <check modifier list> ::= <check modifier>|                                132
                              <check modifier list> <separator>                107 82
                              <check modifier>                                 132

108 <iolength modifier> ::= IOLENGTH  = <primary> <iolength units>             52  133

109 <read CR> ::= READ <unit specifier> <read CR modifier list>                101 134
                  <I/O error disposition>                                      65

110 <read DK> ::= READ <unit or file specifier> EBCDIC                         99
                  <read DK modifier list> <I/O error disposition>              135 65

111 <read MT> ::= READ <unit specifier> <read MT modifier list>                101 136
                  <I/O error disposition>                                      65

112 <read PR> ::= READ <unit specifier> <read PR modifier list>                101 137
                  <I/O error disposition>                                      65

113 <read SC> ::= READ <unit specifier> EBCDIC <read SC modifier list>         101 138
                  <I/O error disposition>                                      65

114 <wait part> ::= <empty> | AND WAIT                                         4

115 <paper motion> ::= <channel modifier>|                                     139
                       <spacing modifier>                                      140
```

```
116 <write CP> ::= WRITE <unit specifier> <write CP modifier list>          101 141
                   <I/O error disposition>                                  65

117 <write DK> ::= WRITE <unit or file specifier> <write DK modifier list>  99 142
                   <I/O error disposition>                                  65

118 <write LP> ::= WRITE <unit specifier> <write LP modifier list>          101 143
                   <I/O error disposition>                                  65

119 <write MT> ::= WRITE <unit specifier> <write MT modifier list>          101 144
                   <I/O error disposition>                                  65

120 <write PP> ::= WRITE <unit specifier> <write PP modifier list>          101 145
                   <I/O error disposition>                                  65

121 <write SC> ::= WRITE <unit specifier> EBCDIC <write SC modifier list>   101 146
                   <I/O error disposition>                                  65

122 <standard I/O modifier> ::= <buffer modifier>|                          147
                                <framesize modifier>|                       148
                                <iolength modifier>|                        108
                                <memory protect modifier>|                  149
                                <pattern modifier>|                         150
                                <result modifier>|                          151
                                <tag modifier>|                             152
                                <translate modifier>                        153

123 <unique I/O modifier> ::= <channel modifier>|                           139
                              <density modifier>|                           154
                              <leader modifier>|                            155
                              <parity modifier>|                            156
                              <segment type/address> <modifier>|            157
                              <spacing modifier>|                           140
                              <stacker modifier>                            158
```

```
124 <I/O error action> ::= <call statement>|                          42
                           CHECK LENGTH|
                           MAST = <primary>|                          52
                           NO RESULT|
                           NOISE|
                           PRINT DATA|
                           PRINT RESULT|
                           SET <variable identifier> = RESULT|        27
                           STOP

125 <disk address part> ::= <empty>|                                  4
                            <absolute disk address>|                  159
                            <relative disk address>                   160

126 <disk test part>  ::= SELECT <disk test list>                     161

127 <tape test list>  ::= <tape test>|                                162
                          <tape test list> <tape test>                127  162

128 <testape modifier list>  ::= <empty>                              4
                                 <testape modifier>                   163
                                 <testape modifier list> <separator>  128  82
                                 <testape modifier>                   163

129 <pattern element>  ::= <repeat part> " <string> "|                164
                           <repeat part> HEX " <hex string> "         164  191
                           <repeat part> <variable identifier> FOR    164  27
                           <field width>|                             165
                           <field width> RIPPLE " <character> "       165  195
                           <increment>|                               79
                           <repeat part> ( <pattern list> )           164  103

130 <compare error action>  ::= <call statement>|                     42
                                PRINT DATA|
                                SET <variable identifier> = RESULT|   27
                                STOP
```

```
131 <single space>  ::= A single unit of blank horizontal spacing

132 <check modifier> ::= <segment type/address modifier>|                157
                         <result modifier>                               151

133 <IOLENGTH UNITS> ::= CHARACTERS|
                         SEGMENTS|
                         WORDS

134 <read CR modifier list> :: <standard I/O modifier>|                  122
                               <read CR modifier list> <separator>       134   82
                               <standard I/O modifier>                   122

135 <read DK modifier list> ::= <read DK modifier>|                      166
                                <read DK modifier list> <separator>      135   82
                                <read DK modifier>                       166

136 <read MT modifier list> ::= <read MT modifier>|                      167
                                <read MT modifier list> <separator>      136   82
                                <read MT modifier>                       167

137 <read PR modifier list> ::= <read PR modifier>|                      168
                                <read PR modifier list> <separator>      137   82
                                <read PR modifier>                       168

138 <read SC modifier list> ::= <read SC modifier>|                      169
                                <read SC modifier list> <separator>      138   82
                                <read SC modifier>                       169
139 <channel modifier> ::= CHANNEL <primary>                             52

140 <spacing modifier> ::= <space amount> SPACE                          170

141 <write CP modifier list> ::= <write CP modifier>|                    171
                                 <write CP modifier list> <separator>    141   82
                                 <write CP modifier>                     171
```

```
142  <write DK modifier list> ::= <write DK modifier>|                    172
                                  <write DK modifier list> <separator>    142  82
                                  <write DK modifier>                     172

143  <write LP modifier list> ::= <write LP modifier>|                    173
                                  <write LP modifier list> <separator>    143  82
                                  <write LP modifier>                     173

144  <write MT modifier list> ::= <write MT modifier>|                    174
                                  <write MT modifier list> <separator>    144  82
                                  <write MT modifier>                     174

145  <write PP modifier list> ::= <write PP modifier>|                    175
                                  <write PP modifier list> <separator>    145  82
                                  <write PP modifier>                     175

146  <write SC modifier list> ::= <write SC modifier>|                    176
                                  <write SC modifier list> <separator>    146  82
                                  <write SC modifier>                     176

147  <buffer modifier> ::= FROM BUFFER <identifier>|                      20
                           INTO BUFFER <identifier>                       20

148  <framesize modifier> ::= BINARY|
                              BCL|
                              EBCDIC

149  <memory protect modifier> ::= MEMORY PROTECT

150  <pattern modifier> ::= <pattern source>                              38

151  <result modifier> ::= SET <variable identifier> = RESULT            27
152  <tag modifier> ::= SET <tag part> TAG                                177
153  <translate modifier> ::= BINARY|
                              TRANSLATE
```

```
154 <density modifier> ::= DENSITY = <primary>                                  52

155 <leader modifier> ::= LEADER

156 <parity modifier> ::= EVEN PARITY|
                          ODD PARITY

157 <segment type/address modifier> ::= <segment type> <segment address>        178 179

158 <stacker modifier> ::= STACKER = <primary>                                  52

159 <absolute disk address> ::= SEGMENT <primary> <ending disk address>         52 180

160 <relative disk address> ::= OFFSET <primary> <area length>                  52 74

161 <disk test list> ::= <disk test>|                                           181
                         <disk test list> <disk test>                          161 181

162 <tape test> ::= ALL|
                    ALL except <test number>|                                   182
                    TEST <test number> <tape test modifier list>                182 183

163 <testape modifier> ::= IGNORE|MONITOR|
                           TRACE <testape trace option>                         184

164 <repeat part> ::= <empty>|                                                  4
                      <number>                                                  8

165 <field width> ::= <empty>|                                                  4
                      <number>                                                  8

166 <read DK modifier> ::= <standard I/O modifier>|                             122
                           <segment type/address modifier>                      157
```


PRINTED IN U.S. AMERICA 5000276

```
167 <read MT modifier> ::= <standard I/O modifier>|                    122
                           <density modifier>|                         154
                           <parity modifier>                           156

168 <read PR modifier> ::= <standard I/O modifier>|                    122
                           <channel modifier>                          139

169 <read SC modifier> ::= <standard I/O modifier>                     122

170 <space amount> ::= DOUBLE|NO|SINGLE

171 <write CP modifier> ::= <standard I/O modifier>|                   122
                            <stacker modifier>                         158

172 <write DK modifier> ::= <standard I/O modifier>|                   122
                            <segment type/address modifier>            157

173 <write LP modifier> ::= <standard I/O modifier>|                   122
                            <channel modifier>|                        139
                            <spacing modifier>                         140

174 <write MT modifier> ::= <standard I/O modifier>|                   122
                            <density modifier>|                        154
                            <parity modifier>                          156

175 <write PP modifier> ::= <standard I/O modifier>|                   122
                            <channel modifier>|                        139
                            <leader modifier>                          155

176 <write SC modifier> ::= <standard I/O modifier>                    122

177 <tag part> ::= CODE|DOUBLE|SINGLE|TRANSFER

178 <segment type> ::= MAINTENANCE|SEGMENT|SEGMENT

179 <segment address> ::= <beginning disk address> <area length>       185 184
```

```
180 <ending disk address> ::= FOR <primary>|                          52
                              THRU <primary>                          52

181 <disk test> ::= ALL|                                              182
                    ALL EXCEPT <test number>|                         182
                    TEST <test number> <disk test modifier list>      182  186

182 <test number> ::= <number>                                        8

183 <tape test modifier list> ::= <empty>|                            4
                                  <tape test modifier>|               187
                                  <tape test modifier list> <separator>  183  82
                                  <tape test modifier>                187

184 <testape trace option> ::= READ|WRITE|ERRORS|ALL

185 <beginning disk address> ::= <primary>|                           52
                                 OFFSET <primary>                     52

186 <disk test modifier list> ::= <empty>|                            4
                                  <disk test modifier>|               188
                                  <disk test modifier list> <separator>  186  82
                                  <disk test modifier>                188

187 <tape test modifier> ::= MONITOR|NO|NOISE|PRINT DATA|PRINT RESULT|
                             REPEAT <number> TIMES|                   8
                             TRACE <tape test trace option>           189

188 <disk test modifier> ::= MONITOR|NO|NOISE|PRINT DATA|PRINT RESULT|
                             REPEAT <number> TIMES|                   8
                             TRACE <disk test trace option>           190

189 <tape test trace option> ::= READ|WRITE|ERRORS|ALL

190 <disk test trace option> ::= READ|WRITE|ERRORS|ALL
```

191 <hex string> ::= <hex digit> | <hex string> <hex digit> 191 192

192 <hex digit> ::= 0|1|2|3|4|5|6|7|8|9|A|B|C|D|E|F

193 <octal string> ::= <octal digit> | <octal string> <octal digit> 193 194

194 <octal digit> ::= 0|1|2|3|4|5|6|7

195 <character> ::= {any CONRAC character except the ETX}

196 <string> ::= {any series of EBCDIC characters excluding quote mark and and ETX}

SYSTEM

MEMORY AND SCAN BUS TEST POINTS

NOTE

All test points are located on backplane of cable driver cards.

Backplane Pins	Address 0 thru 19		Information 0 thru 19		Information 20 thru 39		Info. 40 thru 51 And Mem. Control		Scan Control	
	X	Y	X	Y	X	Y	X	Y	X	Y
1X			00D	10D	20D	30D	40D	50D		MXBC
0V			01D	11D	21D	31D	41D			
1U			02D	12D	22D	32D	42D			EIMB
0R			03D	13D	23D	33D	43D			
1N			04D	14D	24D	34D	44D	51D		SIPL
1B			05D	15D	25D	35D	45D	MAOn		SAOX
1D			06D	16D	26D	36D	46D	MRY		SRDY
1F			07D	17D	27D	37D	47D			MXSD
1H			08D	18D	28D	38D	48D	MAB		EIMA
1L			09D	19D	29D	39D	49D	MTE		STEF
1Y	10R	00R	00R	10R	20R	30R	40R	50R		
1W	01R	11R	01R	11R	21R	31R	41R	REQn		SREQ
1V	02R	12R	02R	12R	22R	32R	42R			
1T	03R	13R	03R	13R	23R	33R	43R	WRC		SWRL
1R	04R	14R	04R	14R	24R	34R	44R	51R		51R
1C	05R	15R	05R	15R	25R	35R	45R			
1E	06R	16R	06R	16R	26R	36R	46R			
0H	07R	17R	07R	17R	27R	37R	47R	PRC		MXSD
0J	08R	18R	08R	18R	28R	38R	48R		LOAX	
0M	09R	19R	09R	19R	29R	39R	49R	ACP	LSEL	SAPL
0L								MRS		
0X	10/D	00/D	00/D	10/D	20/D	30/D	40/D			
0W	01/D	11/D	01/D	11/D	21/D	31/D	41/D			SREQ/
0U	02/D	12/D	02/D	12/D	22/D	32/D	42/D			
0S	03/D	13/D	03/D	13/D	23/D	33/D	43/D	WRC/		SWRL/
1P	04/D	14/D	04/D	14/D	24/D	34/D	44/D			
0C	05/D	15/D	05/D	15/D	25/D	35/D	45/D			
0E	06/D	16/D	06/D	16/D	26/D	36/D	46/D	MRY/		SRDY/
1G	07/D	17/D.	07/D	17/D	27/D	37/D	47/D	PRC/		MXSD/
1J	08/D	18/D	08/D	18/D	28/D	38/D	48/D			
1M	09/D	19/D	09/D	19/D	29/D	39/D	49/D			SAPL/

X = Upper half of card
Y = Lower half of card
R = Receiver (from bus)
D = Driver (to bus)

SCAN BUS CARD LOCATIONS

Unit	Address	Information 00 thru 19	Information 20 thru 39	Information 40 thru 51 and Control
Processor	ACDI4	ACDC2	ACDE2	ACDG4
Multiplexor	DBDJ2	DBDI3	DBDI6	DBDI9
Data Comm	FBDF2	FBDE9	FBDE6	FBDE3

MEMORY INTERFACE HUB CARD LOCATIONS

Unit	Hub	Address	Information 00 thru 19	Information 20 thru 39	Information 40 thru 51 and Control
Memory Tester (EAD= bottom of card)	A	EADA5	EADC2	EADD7	EADF4
	B	EADA8	EADC5	EADE0	EADF7
	C	EADB1	EADC8	EADE3	EADG0
	D	EADB4	EADD1	EADE6	EADG3
	E	EADB9	EADD4	EADE9	EADG6
Memory Cabinet (CBD= bottom of card)	A	CBDB5	CBDH8	CBDF7	CBDD6
	B	CBDB2	CBDH5	CBDF4	CBDD3
	C	CBDA9	CBDH2	CBDF1	CBDD0
	D	CBDA6	CBDG9	CBDE8	CBDC7
	E	CBDA3	CBDG6	CBDE5	CBDC4
	F	CBDA0	CBDG3	CBDE2	CBDC1
Multi-plexor (DAB=top of card)	A	DABA5	DABC2	DABD7	DABF4
	B	DABA8	DABC5	DABE0	DABF7
	C	DABB1	DABC8	DABE3	DABG0
	D	DABB4	DABD1	DABE6	DABG3
	E	DABB9	DABD4	DABE9	DABG6
Processor (ACD=top of card)	A	ACDG7	ACDA5	ACDC7	ACDE9
	B	ACDH0	ACDA8	ACDD0	ACDF2
	C	ACDH5	ACDB1	ACDD3	ACDF5
	D	ACDH8	ACDB4	ACDD6	ACDF8
	E	ACDI1	ACDB9	ACDD9	ACDG1
MDL (DDD= bottom of card)	A	DDDF0	Not Used	DDDE4	DDDE1

PROCESSOR

BACKPLANE CLOCK TEST POINTS

PANEL A

Signal	End of Net T.P.
CLKA.A01	AABA2B
CLKA.A02	AABC7F
CLKA.A05	AADB7D
CLKA.A07	AAFA6N
CLKA.A04	AADE6D
CLKA.A11	AAFB1D
CLKA.A09	AAFB7D
CLKA.A08	AAFE5D
CLKA.A12	AAHB7D
CLKB.B01	AABJ9D
CLKB.B02	AADG6D
CLKA.B06	AADG7B
CLKA.B03	AADJ3D
CLKB.B04	AADJ5D
CLKB.B05	AADJ9D
CLKB.B12	AAFJ6C
CLKA.B07	AAFK1P
CLKB.C03	AAFK0P
CLKA.C01	AAHH0C
CLKA.C02	AAHJ3D
CLKA.C05	AAHK2C
CLKA.C04	AAJE5D
CLKA.C08	AAJH8D
CLKA.C12	AALF1D

PANEL B

Signal	End of Net T.P.
CLKB.I12	ABDB2X
CLKA.I12	ABFA6B
CLKA.I11	ABFA8B
CLKA.I10	ABFB0B
CLKA.I09	ABFB2B
CLKA.I08	ABFB4B
CLKA.I07	ABFH6B
CLKB/J05	ABBI2S
CLKB/J08	ABBJ2Y
CLKB/J11	ABBK2Y
CLKB.J07	ABDJ6B
CLKB.J09	ABDJ8B
CLKB.J10	ABDK0B

BACKPLANE CLOCK TEST POINTS (Cont)

PANEL B (Cont)

Signal	End of Net T.P.
CLKB.J12	ABDK2B
CLKA.J12	ABFE8B
CLKA.J11	ABFF0B
CLKA.J10	ABFF2B
CLKA.J09	ABFF4B
CLKA.J06	ABFK0B
CLKA.J05	ABFK2B
CLKA.J07	ABFK4B
CLKA.J08	ABFK6B
CLKA.K01	ABJC8F
CLKA.K09	ABLC2F
CLKA.K12	ADLD3D
CLKB/K02	ABLI6B
CLKB/K03	ABLI6C
CLKA.K04	ABLI6D

PANEL C

Signal	End of Net T.P.
CLKA.M03	ACFJ6Y
CLKA.M04	ACFK0Y
CLKA.M02	ACFK2Y
CLKA.M01	ACHB4V
CLKA.M07	ACHF4B
CLKA.M08	ACHF6B
CLKA.M05	ACHJ4E
CLKA.N02	ACJB4F
CLKA.N01	ACJD2B
CLKA.N04	ACJE2B
CLKA.N05	ACJG0B
CLKA.N11	ACLF6C
CLKA.N09	ACLG2B
CLKA.N12	ACLH2B

PROCESSOR

MEMORY AND SCAN BUS OUTPUT TEST POINTS

Information Bits

00 ACDE6X	18 ACFE6H	36 ACFE7D
01 ACDE6V	19 ACFE6L	37 ACFE7F
02 ACDE6U	20 ACDE7X	38 ACFE7H
03 ACDE6R	21 ACDE7V	39 ACFE7L
04 ACDE6N	22 ACDE7U	40 ACDE6Y
05 ACDE6B	23 ACDE7R	41 ACDE6W
06 ACDE6D	24 ACDE7P	42 ACDE6T
07 ACDE6F	25 ACDE7C	43 ACDE6S
08 ACDE6H	26 ACDE7D	44 ACDE6P
09 ACDE6L	27 ACDE7F	45 ACDE6C
10 ACFE6X	28 ACDE7H	46 ACDE6E
11 ACFE6V	29 ACDE7L	47 ACDE6G
12 ACFE6U	30 ACFE7X	48 ACDE6I
13 ACFE6R	31 ACFE7V	49 ACDE6M
14 ACFE6N	32 ACFE7U	50 ACFE7W
15 ACFE6B	33 ACFE7R	51 ACFE7M
16 ACFE6D	34 ACFE7P	
17 ACFE6F	35 ACFE7C	

Address Bits

00 ACFE6Y	10 ACDE7W
01 ACDE7S	11 ACFE6W
02 ACDE7Q	12 ACFE6T
03 ACDE7M	13 ACFE6S
04 ACDE7K	14 ACFE6P
05 ACDE7E	15 ACFE6C
06 ACDE7I	16 ACFE6E
07 ACDE7J	17 ACFE6G
08 ACDE6J	18 ACFE6I
09 ACDE6K	19 ACFE6M

Control

MPRC.RM1	ACFE7E
MAPL.LM1	ACFE7K
MWRC.LM1	ACFE7Q
MREQ.RM1	ACFE7S
SCAN.CXC	ACDG5S
LOAX.B11	ACDG5H
LSEL.B11	ACDG5L

MEMORY AND SCAN BUS INPUT TEST POINTS

Information Bits

00 ACDA6Y	18 ACFA5J	36 ACFC8E
01 ACDA6W	19 ACFA5M	37 ACFC8H
02 ACDA6V	20 ACDC8Y	38 ACFC7J
03 ACDA6T	21 ACDC8W	39 ACFC7M
04 ACDA6R	22 ACDC8V	40 ACDF0Y
05 ACDA6C	23 ACDC8T	41 ACDF0W
06 ACDA6E	24 ACDC8R	42 ACDF0V
07 ACDA5H	25 ACDC8C	43 ACDF0T
08 ACDA5J	26 ACDC8E	44 ACDF0R
09 ACDA5M	27 ACDC7H	45 ACDF0C
10 ACFA6Y	28 ACDC7J	46 ACDF0E
11 ACFA6W	29 ACDC7M	47 ACDE9H
12 ACFA6V	30 ACFC8Y	48 ACDE9J
13 ACFA6T	31 ACFC8W	49 ACDE9M
14 ACFA6R	32 ACFC8V	50 ACFF0Y
15 ACFA6C	33 ACFC8T	51 ACFF0R
16 ACFA6E	34 ACFC8R	
17 ACFA5H	35 ACFC8C	

Control

MAOX.C00	ACFF0C
MABX.C00	ACFD9J
MTEX.C00	ACFE9M
MRDY.C00	ACFF0E
SRDY/X00	ACFG4E
EIMA.X00	ACFG4J
EIMB.X00	ACFG5V
MXSD.X00	ACFG4H
SAOX.X00	ACFG5C
STEX.X00	ACFG4M

LOGIC VOLTAGE TEST POINTS

NOTE

The following test points are end-of-net points; therefore, a proper indication at any of these points signifies that the associated voltage net is good.

+ 4.75V	- 2.0V
AABK9A	AABK9Z
AADK9A	AADK9Z
AAFK9A	AAFK9Z
AAHK9A	AAHK9Z
AAJK9A	AAJK9Z
AALK9A	AALK9Z

PROCESSOR

LOGIC VOLTAGE TEST POINTS (Cont)

+ 4.75V	- 2.0V
ABBK9A	ABBK9Z
ABDK9A	ABDK9Z
ABFK9A	ABFK9Z
ABHK9A	ABHK9Z
ABJK9A	ABJK9Z
ABLK9A	ABLK9Z
ACBK9A	ACBK9Z
ACDK9A	ACDK9Z
ACFK9A	ACFK9Z
ACHK9A	ACHK9Z
ACJK9A	ACJK9Z
ACLK9A	ACLK9Z

SPECIAL VOLTAGE TEST POINTS

+ 1V	+ 12V	- 12V	- 4.5V
To Cable Driver/Receiver Cards:			
ACDA51	ACDA5K	ACFA51	
ACDF31	ACDF3K	ACFF31	
ACD171	ACD17K	ACF171	
To Clock Circuit Cards:			
	AADC6K	AABD0C	
	AADH1K		
	AAJ17K		
	ABBD0K		
	ABBH5K		
	ABDD0K		
	ABDH5K		
	ABLE1K		
	ACHH2K		
	ACJF5K		

NOTE

+1 volt is applied to all Clock Distribution cards. Test points for Type I and Type IV cards are pins 1C, 1D, 1M, 1N, and 1P. For Type II cards, test points are pins 0B and 0C.

INTERFACE CABLES

	Output Line	Memory Bus	Scan Bus	Description
Cable 1:	W	MA00	SA00	
	T	MA01	SA01	
	R	MA02	SA02	
	P	MA03	SA03	
	M	MA04	SA04	
	C	MA05	SA05	
	E	MA06	SA06	
	G	MA07	SA07	
	I	MA08	SA08	
	K	MA09	SA09	Memory/Scan
	V	MA10	SA10	Address Bits
	U	MA11	SA11	00 thru 19
	S	MA12	SA12	
	Q	MA13	SA13	
	N	MA14	SA14	
	D	MA15	SA15	
	F	MA16	SA16	
	H	MA17	SA17	
	J	MA18	SA18	
	L	MA19	SA19	
Cable 2:	V	MI00	SI00	
	T	MI01	SI01	
	R	MI02	SI02	
	P	MI03	SI03	
	M	MI04	SI04	
	C	MI05	SI05	
	E	MI06	SI06	
	G	MI07	SI07	
	I	MI08	SI08	
	K	MI09	SI09	Memory/Scan
	W	MI10	SI10	Information Bits
	U	MI11	SI11	00 thru 19
	S	MI12	SI12	
	Q	MI13	SI13	
	N	MI14	SI14	
	D	MI15	SI15	
	F	MI16	SI16	
	H	MI17	SI17	
	J	MI18	SI18	
	L	MI19	SI19	

PROCESSOR

INTERFACE CABLES (Cont)

	Output Line	Memory Bus	Scan Bus	Description
Cable 3:	V	M120	S120	
	T	M121	S121	
	R	M122	S122	
	P	M123	S123	
	M	M124	S124	
	C	M125	S125	
	E	M126	S126	
	G	M127	S127	
	I	M128	S128	Memory/Scan
	K	M129	S129	Information Bits
	W	M130	S130	20 thru 39
	U	M131	S131	
	S	M132	S132	
	Q	M133	S133	
	N	M134	S134	
	D	M135	S135	
	F	M136	S136	
	H	M137	S137	
	J	M138	S138	
	L	M139	S139	
Cable 4:	V	M140	S140	
	T	M141	S141	
	R	M142	S142	
	P	M143	S143	
	M	M144	S144	Memory/Scan
	C	M145	S145	Information Bits
	E	M146	S146	40 thru 51
	G	M147	S147	
	I	M148	S148	
	K	M149	S149	
	W	M150	S150	
	N	M151	S151	
	F	(1) MRDY	(2) SRDY	(1) Memory Ready (2) Scan Ready
	U	(1) MREQ	(2) SREQ	(1) Memory Request (2) Scan Request
	Q	(1) MWRC	(2) SWRC	(1) Memory Write Control (2) Scan Write Control
	H	(1) MPRC	(2) MXSD	(1) Memory Protect Control (2) MPX-D Ready Line
	J	(1) MABX	(2) EIMA	(1) Memory Access Begun (2) External Interrupt-MPXA
	D	(1) MAOX	(2) SAOX	(1) Memory Access Obtained (2) Scan Access Obtained

INTERFACE CABLES (Cont)

	Output Line	Memory Bus	Scan Bus	Description
Cable 4: (Cont)	S	(1) spare	(2) EIMB	(1) spare (2) External Interrupt-MPXB
	L	(1) MAPL	(3) SAPL	(1) Memory Address Parity Level
		(2) MTEX	(4) STEX	(2) Memory Detected Transmission Error (3) Scan Address Parity Level (4) Scan Transmission Error

IC MEMORY REGISTER ADDRESSES

ADDRESS DEC	ADDRESS HEX	NAME	REGISTER USAGE	LOGIC NAME
0 - 31	00 - 1F	D0-D31	DISPLAYS	
32	20	PIR	PROGRAM INDEX	I x S0
33	21	SIR	SOURCE INDEX	I x S1
34	22	DIR	DESTINATION INDEX	I x S2
35	23	TIR	TABLE INDEX	I x S3
		BUF3	TEMPORARY STORAGE	
36	24	LØSR	LIMIT OF STACK	I x S4
37	25	BØSR	BASE OF STACK	I x S5
38	26	F	MOST RECENT MSCW ADDRESS	I x S6
39	27	BUF	TEMPORARY STORAGE	I x S7
48	30	PBR	PROGRAM BASE	B x S0
49	31	SBR	SOURCE BASE	B x S1
50	32	DBR	DESTINATION BASE	B x S2
51	33	TBR	TABLE BASE	B x S3
		BUF2	TEMPORARY STORAGE	
52	34	S	TOP-OF-STACK ADDRESS	B x S4
53	35	SNR	STACK VECTOR INDEX OF CURRENT STACK	B x S5
54	36	PDR	ADDRESS OF CURRENT SEGMENT DESCRIPTOR	B x S6
55	37	TEMP	TEMPORARY STORAGE	B x S7

x = WRITE/READ (W/R)

CORE MEMORY

POWER SUPPLY VOLTAGE CHECK AND ADJUSTMENT

Test Point	Adjust For	Card Location	Control Location
		NOTE Prior to performing any adjustment, ensure that +12 V, -12 V, and -22 V outputs of memory control cabinet are within 3 per cent of nominal, and that the +10 V and -10 V supplies of the AC Power Module are normal.	
-50V	-50V	Panel D 789	B0
-2V	-2V	Panel B 0	
+1V	+1V	Panel B 0	
+4.5V	+4.5V	Panel A	
+63V	+63V	Panel D 345	G1
+37V	+37V	Panel D 345	J1

MEMORY REGULATORS THRESHOLD VOLTAGE CHECK AND ADJUSTMENT

NOTE

Designations "L" and "R" on special voltage test panel identify left and right stacks, respectively, as viewed from card side. Left stack contains high-order half of all words for associated module.

Test Point	Adjust For	Location	Module	Stack
50L, 43L, LT	Recorded nominal value	Panel E1	L	Left
		Panel E3	M	Left
		Panel D1	N	Left
50R, 43R, RT	Recorded nominal value	Panel E2	L	Right
		Panel E4	M	Right
		Panel D2	N	Right

SENSE AMPLIFIER TEST POINTS

Signal	Circuit 0	Circuit 1	Circuit 2
Strobe	1C	1K	0U
Reset	0D	1N	1V
Output	1D	1L	1U
Test Point	1F	1Q	0X
Threshold	1G	0T	1X

QUADRANT SELECT TEST POINTS

Switch	Pin
LL Sw Q4	0J
LL Sw Q1	1S
LL Sw Q2	1I
LL Sw Q3	0S
Dummy Sw	1J

(Test points are located on Sense Amplifier cards.)

GROUND SWITCH SELECT LEVEL TEST POINTS

Switch Number	Stack 0		Stack 1	
	X	Y	X	Y
0	ADE6D	ADI2D	ADA0D	ADD6D
1	AHE5D	ADI1D	AHA0D	AHD6D
2	ADE7D	ADI3D	ADA1D	ADD7D
3	AHE6D	AHI2D	AHA1D	AHD7D
4	ADE6C	ADI2C	ADA0C	ADD6C
5	AHE5C	ADI1C	AHA0C	AHD6C
6	ADE7C	ADI3C	ADA1C	ADD7C
7	AHE6C	AHI2C	AHA1C	AHD7C
8	ADE7X	ADI3X	ADA1C	ADD7C
9	AHE6X	ADI2X	AHA1X	AHD7X
10	ADE6X	ADI2X	ADA0X	ADD6X
11	AHE5X	AHI1X	AHA0X	AHD6X
12	ADE7W	ADI3W	ADA1W	ADD7W
13	AHE6W	AHI2W	AHA1W	AHD7W
14	ADE6W	ADI2W	ADA0W	ADD6W
15	AHE5W	AHI1W	AHA0W	AHD6W

CORE MEMORY

INHIBIT DRIVER TEST POINTS

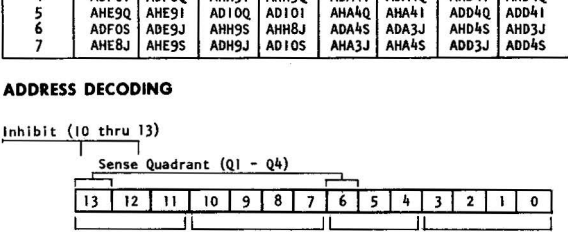
	I0	I1	I2	I3
Input 0	1C	0C	1D	0D
Output 0	0H	1H	0G	1G
Input 1	1J	0J	1K	0K
Output 1	0Q	1Q	0P	1P
Input 2	1S	0S	1T	0T
Output 2	0X	1X	0W	1W

(Inputs are False to enable drivers.)

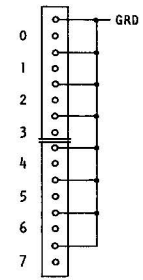
FLOATING SWITCH SELECT LEVEL TEST POINTS

SWITCH NUMBER	STACK 0				STACK 1			
	X		Y		X		Y	
	WRITE	READ	WRITE	READ	WRITE	READ	WRITE	READ
0	ADE9I	ADE9C	AHH8I	AHH8C	ADA3I	ADA3C	AHD3I	AHD3C
1	AHE8C	AHE8I	ADH9C	ADH9I	AHA3C	AHA3I	ADD3C	ADD3I
2	ADF0J	ADF0C	AHH9J	ADD9C	ADA4J	ADA4C	AHD4J	AHD4C
3	AHE9C	AHE9J	AD10C	AD10J	AHA4C	AHA4J	ADD4C	ADD4J
4	ADF0I	ADF0Q	AHH9I	AHH9Q	ADA4I	ADA4Q	AHD4I	AHD4Q
5	AHE9Q	AHE9I	AD10Q	AD10I	AHA4Q	AHA4I	ADD4Q	ADD4I
6	ADF0S	ABE9J	AHH9S	AHH8J	ADA4S	ADA3J	AHD4S	AHD3J
7	AHE8J	AHE9S	ADH9J	AD10S	AHA3J	AHA4S	ADD3J	ADD4S

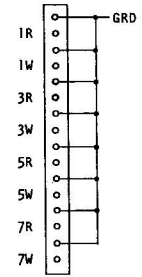
ADDRESS DECODING



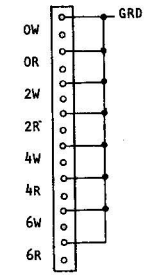
FLOATING AND GROUND SWITCH CARD CONNECTIONS



Ground Switch Connector



Floating Switch Connector ODD



Floating Switch Connector EVEN

TIMING GENERATOR OUTPUTS

Name	Timing Tab	Circuit	Output Pin
MTGA.D00	0	1	1H
MTGB.D00	345	2	1E
MTGC.D00	30	5	0J
MTGD.D00	110	3	1G
MTGE.D00	40	4	0H
MTGJ.D00	140	6	1K
*MTGM.D00	235	7	1R
MTGQ.D00	265	9	0W
*MTGR.D01	295	11	1V
MTGS.D00	435	12	1U
MTGT.D00	375	8	1T
MTGU.D00	380	13	0Q
MTGV.D00	410	14	1N
MTGW.D00	450	15	1J
MTGY.D00	420	17	1C
MTGZ.D00	450	16	0M
MTGF.D00	280	19	1M
*MTGM.D01	235	20	1Q
*MTGR.D00	295	21	0T

* Tab varies per stack.

(Input: DLGD.L00, Pin 0C)

CORE MEMORY

MEMORY MODULE TIMING SIGNALS

NOTE

All times are referenced to MTGA output (Tab 0) from Timing Generator delay line.

Signal	Test Point	Function	Leading	Trailing	Waveform Parameters
DLGD.L00	CAJG0C	Input to delay line driver	MTGD0R00 (BUSY flip-flop)	MTGY	-30ns 420ns
STRB.B00 STRB.B01	CAJG3N CAJB9N	Read Current enable	MTGA	MTGQ	0ns 265ns
STRB.B10 STRB.B11	CAJG2K CAJB8K	Write Current enable	MTGW	MTGF/	450ns 730ns
ST03 ST04		Sense amplifier strobe	MTGM (CAJG1R, CAJG1Q)	MTGR (CAJG0T, CAJG1V)	235ns 295ns
ST07.L00	CAJG4G	Enable Read floating switch	MTGD1R00 (BUSY flip-flop)	MTGT	-30ns 375ns
ST08.L00	CAJB8F	Enable write floating switch	MTGV	MTGB/	410ns 795ns
ST09.L00	CAJG5M	Enable ground switch	MTGD1R00 (BUSY flip-flop)	MTGT/	-30ns 825ns
ST10.L00	CAJG3F	Write Strobe into MWR	MTGD	MTGJ	110ns 140ns
ST05.L00	CAJG2E	Inhibit driver enable	MTGS	MTGU/	435ns 830ns
ST02.000	CAJG3S	SA quadrant select enable	MTGA	MTGU	0ns 380ns
Signal	Test Point	Function	Leading	Trailing	Waveshape
RESW.B00 RESW.B10	CAJG2M CAJB8M	Restore	MTGV/	MTGZ/	860ns 900ns
SETA.L00	CAJG4S	Set address enable	BUSY/	MTGV/	-30ns 860ns

MULTIPLEXOR

MEMORY AND SCAN BUS HUB CARD LOCATIONS

MEMORY INTERFACE				
Memory Hub	Address	Information 00 thru 19	Information 20 thru 39	Information 40 thru 51
HUB A	DABA5	DABC2	DABD7	DABF4
HUB B	DABA8	DABC5	DABE0	DABF7
HUB C	DABB1	DABC8	DABE3	DABG0
HUB D	DABB4	DABD1	DABE6	DABG3
HUB E	DABB9	DABD4	DABE9	DABG6
(DAB = Top of Card)				
SCAN BUS INTERFACE				
Addresses Bits 00 thru 19 Bits 20 thru 39 Bits 40 thru 51 and Controls		DBDJ2 DBDI3 DBDI6 DBDI9		
(DBD = Bottom of Card)				

GENERAL PROCEDURE FOR MANUAL CARD TEST

1. Set LOCAL REMOTE switch to LOCAL.
2. Set CARD TEST ENABLE OFF switch to ENABLE.
3. Clear TIME OF DAY register.
4. Set desired test case bits into TIME OF DAY register and insert card to be tested into location DBBD8. Correspondence of register bits and backplane pins is indicated in the following chart.

MULTIPLEXOR

GENERAL PROCEDURE FOR MANUAL CARD TEST (Cont)

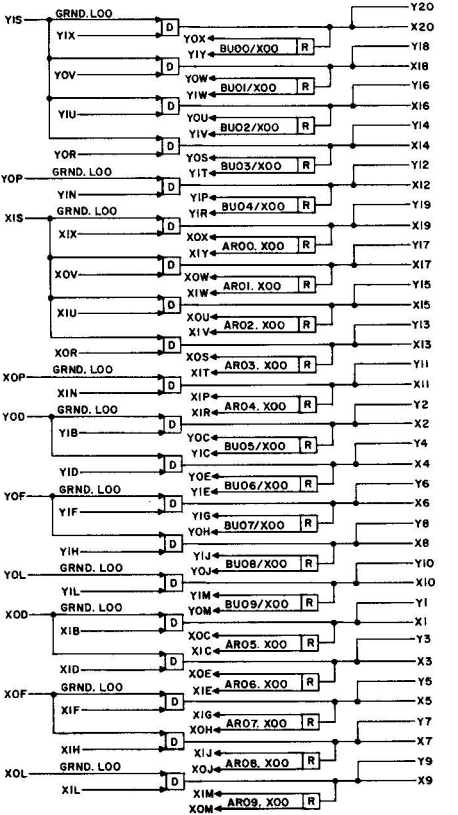
TIME OF DAY Register Bit	Backplane Pin
TR00.F00	0B
TR01.F00	0C
TR02.F00	0D
TR03.F00	0E
TR04.F00	0F
TR05.F00	0G
TR06.F00	0H
TR07.F00	0I
TR08.F00	0J
TR09.F00	0K
TR10.F00	0L
TR11.F00	0M
TR12.F00	0N
TR13.F00	0P
TR14.F00	0Q
TR15.F00	0R
TR16.F00	0S
TR17.F00	0T
TR18.F00	0U
TR19.F00	0V
TR20.F00	0W
TR21.F00	0X
TR22.F00	0Y

TIME OF DAY Register Bit	Backplane Pin
TR23.F00	1C
TR24.F00	1D
TR25.F00	1E
TR26.F00	1F
TR27.F00	1G
TR28.F00	1H
TR29.F00	1I
TR30.F00	1J
TR31.F00	1K
TR32.F00	1L
TR33.F00	1M
TR34.F00	1N
TR35.F00	1P
TR36.F00	1Q
TR37.F00	1R
TR38.F00	1S
TR39.F00	1T
TR40.F00	1U
TR41.F00	1V
TR42.F00	1W
TR43.F00	1X

PERIPHERAL CONTROLLER INTERFACE

Location DBJJ5 (PCC 0)
DBJI9 (PCC 1)

Busy and Access Request (ARL)

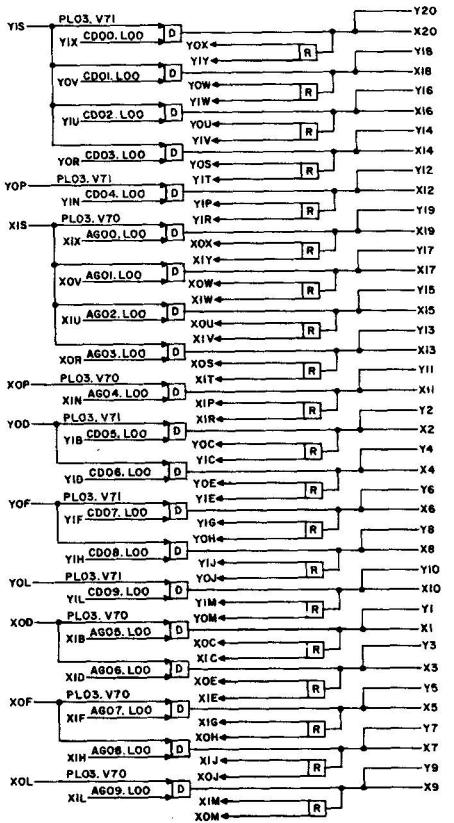


MULTIPLEXOR

PERIPHERAL CONTROLLER INTERFACE (Cont)

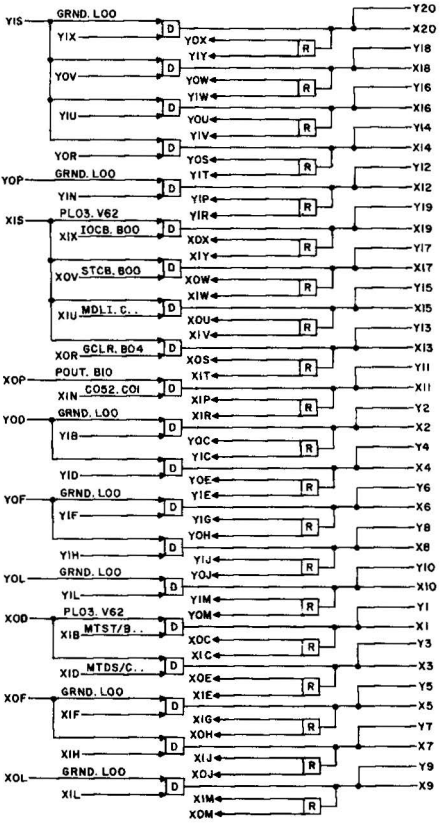
Access Granted (AGL) and Channel Designate (CDL)

Location DBJJ8 (PCC 0)
DBJJ2 (PCC 1)



PERIPHERAL CONTROLLER INTERFACE (Cont)

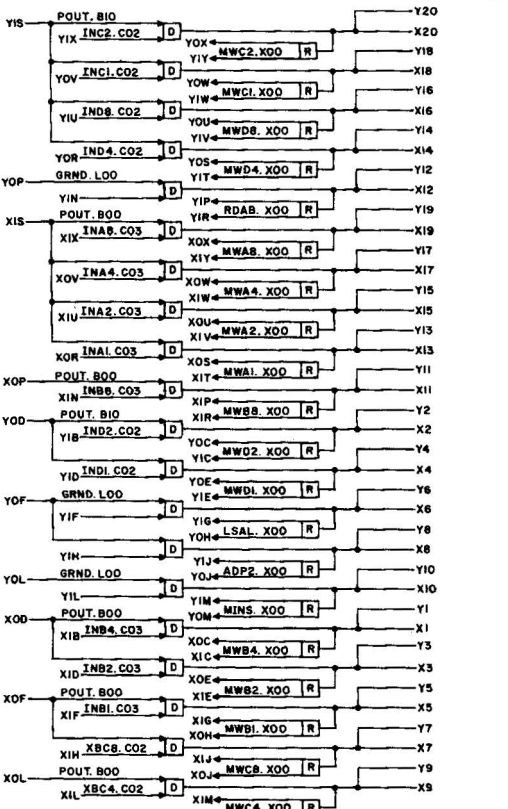
Control Signals Location DBJ16



MULTIPLEXOR

PERIPHERAL CONTROLLER INTERFACE (Cont)

Information and Control Location DBJ13



DISPLAY AND MDL

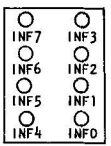
Multiplexor and Processor Bit Assignments (Panel B)

MULTIPLEXOR

DEC	119	118	057	056	116	117	064	063	062	061
OCT	167	166	071	070	164	165	100	077	076	075
LOC	HB3W	HB3T	HB2S	HB2P	HB3Q	HB2D	HB2F	HB3G	HB3J	HB2J
BIT	0952	0944	0456	0448	0928	0936	0512	0504	0496	0488
DEC	127	126	124	125	123	120	121	122		
OCT	177	176	174	175	173	170	171	172	Spare	Spare
LOC	HB0W	HB0T	HA9S	HA9P	HB0Q	HA9D	HA9F	HB0G		
BIT	1016	1008	0992	1000	0984	0960	0968	0976		
DEC	010	009	008	007	006	005	004	003	002	001
OCT	012	011	010	007	006	005	004	003	002	001
LOC	HA7L	HA6L	HA6X	HA7X	HA6V	HA7W	HA7T	HA6S	HA6P	HA7Q
BIT	0080	0072	0064	0056	0048	0040	0032	0024	0016	0008
DEC	020	019	018	017	016	015	014	013	012	011
OCT	024	023	022	021	020	017	016	015	014	013
LOC	HA6D	HA6F	HA7G	HA7J	HA6J	HA4L	HA3L	HA3X	HA4X	HA3V
BIT	0160	0152	0144	0136	0128	0120	0112	0104	0096	0088
DEC	033	032	031	030	029	028	027	026	025	024
OCT	041	040	037	036	035	034	033	032	031	030
LOC	HA4W	HA4T	HA3G	HA3P	HA4Q	HA3D	HA3F	HA4G	HA4J	HA3J
BIT	0264	0256	0248	0240	0232	0224	0216	0208	0200	0192
DEC	040	023	022	021	039	054	052	051	035	034
OCT	050	027	026	025	047	066	064	063	043	042
LOC	HA1L	HA0L	HA0X	HA1X	HA0V	HA1W	HA1T	HA0S	HA0P	HA1Q
BIT	0320	0184	0176	0168	0312	0432	0416	0408	0280	0272
DEC	037	036	048	047	046	045	044	043	042	041
OCT	045	044	060	057	056	055	054	053	052	051
LOC	HA0D	HA0F	HA1G	HA1J	HA0J	JB3L	JB2L	JB2X	JB3X	JB2V
BIT	0296	0288	0384	0376	0368	0360	0352	0344	0336	0328
DEC	060	059	058			055	050	049	053	038
OCT	074	073	072	Spare	Spare	067	062	061	065	046
LOC	JB3W	JB3T	JB2S			JB2D	JB2F	JB3G	JB3J	JB2J
BIT	0480	0472	0464			0440	0400	0392	0424	0304

NOTE

DEC - Decimal word address
OCT - Octal word address
LOC - Pin location on rack D, panel D, for a scope sync pulse for that particular word
BIT - Address of the first, or zero bit of each address and is derived by multiplying the decimal word address by eight. This bit address occurs in the MDL basic listing.



DISPLAY AND MDL

PROCESSOR

DEC	247	246	185	184	244	245	192	191	190	189
OCT	367	366	271	270	364	365	300	277	276	275
LOC	HB3W	HB3T	HB2S	HB2P	HB3Q	HB2D	HB2F	HB3G	HB3J	HB2J
BIT	1976	1968	1480	1472	1952	1960	1536	1528	1520	1512
DEC	255	254	252	253	251	248	249	250		
OCT	377	376	374	375	373	370	371	372	Spare	Spare
LOC	HB0W	HB0T	HA9S	HA9P	HB0Q	HA9D	HA9F	HB0G		
BIT	2040	2032	2016	2024	2008	1984	1992	2000		
DEC	138	137	136	135	134	133	132	131	130	129
OCT	212	211	210	207	206	205	204	203	202	201
LOC	HA7L	HA6L	HA6X	HA7X	HA6V	HA7W	HA7T	HA6S	HA6P	HA7Q
BIT	1104	1096	1088	1080	1072	1064	1056	1048	1040	1032
DEC	148	147	146	145	144	143	142	141	140	139
OCT	224	223	222	221	220	217	216	215	214	213
LOC	HA6D	HA6F	HA7G	HA7J	HA6J	HA4L	HA3L	HA3X	HA4X	HA3V
BIT	1184	1176	1168	1160	1152	1144	1136	1128	1120	1112
DEC	161	160	159	158	157	156	155	154	153	152
OCT	241	240	237	236	235	234	233	232	231	230
LOC	HA4W	HA4T	HA3G	HA3P	HA4Q	HA3D	HA3F	HA4G	HA4J	HA3J
BIT	1288	1280	1272	1264	1256	1248	1240	1232	1224	1216
DEC	168	151	150	149	167	182	180	179	163	162
OCT	250	227	226	225	247	266	264	263	243	242
LOC	HA1L	HA0L	HA0X	HA1X	HA0V	HA1W	HA1T	HA0S	HA0P	HA1Q
BIT	1344	1208	1200	1192	1336	1456	1440	1432	1304	1296
DEC	165	164	176	175	174	173	172	171	170	169
OCT	245	244	260	257	256	255	254	253	252	251
LOC	HA0D	HA0F	HA1G	HA1J	HA0J	JB3L	JB2L	JB2X	JB3X	JB2V
BIT	1320	1312	1408	1400	1392	1384	1376	1368	1360	1352
DEC	188	187	186			183	178	177	181	166
OCT	274	273	272	Spare	Spare	267	262	261	265	246
LOC	JB3W	JB3T	JB2S			JB2D	JB2F	JB3G	JB3J	JB2J
BIT	1504	1496	1488			1464	1424	1416	1448	1328

O INF7	O INF3
O INF6	O INF2
O INF5	O INF1
O INF4	O INF0

INF0 - DDFB8T
INF1 - DDFB9N
INF2 - DDFB8M
INF3 - DDFB8W
INF4 - DDFB8X
INF5 - DDFB9W
INF6 - DDFB8Y
INF7 - DDFB8U

(See Note on page 8-45.)

Registers (Panel A)

BIT LOC		47	43	39	35	31	27	23	19	15	11	07	03
DEC		240	236	232	228	224	220	216	212	208	204	200	196
OCT		360	354	350	344	340	334	330	324	320	314	310	304
LOC		JA0P	JA0V	JA1L	JA3F	JA3S	JA4X	JA6J	JA6D	JA7T	JA6X	JA9J	JB0X
BIT		1920	1888	1856	1824	1792	1760	1728	1696	1664	1632	1600	1568
BIT LOC	50	46	42	38	34	30	26	22	18	14	10	06	02
DEC	243	239	235	231	227	223	219	215	211	207	203	199	195
OCT	363	357	353	347	343	337	333	327	323	317	313	307	303
LOC	JA0F	JA0S	JA1X	JA3J	JA3D	JA4T	JA3X	JA7J	JA7Q	JA7W	JA6L	JB0T	JA9X
BIT	1944	1912	1880	1848	1816	1784	1752	1720	1688	1656	1624	1592	1560
BIT LOC	49	45	41	37	33	29	25	21	17	13	09	05	01
DEC	242	238	234	230	226	222	218	214	210	206	202	198	194
OCT	362	356	352	346	342	336	332	326	322	316	312	306	302
LOC	JA0D	JA1T	JA0X	JA4J	JA4Q	JA4W	JA3L	JA7G	JA6P	JA6V	JA7L	JB0W	JA9L
BIT	1936	1904	1872	1880	1808	1776	1744	1712	1680	1648	1616	1584	1552
BIT LOC	48	44	40	36	32	28	24	20	16	12	08	04	00
DEC	241	237	233	229	225	221	217	213	209	205	201	197	193
OCT	361	355	351	345	341	335	331	325	321	315	311	305	301
LOC	JA1Q	JA1W	JA0L	JA4G	JA3P	JA3V	JA4L	JA6F	JA6S	JA7X	JA9P	JA9V	JB06
BIT	1928	1896	1864	1832	1800	1768	1736	1704	1672	1640	1608	1576	1544

P REG - INF0 - DDFB8T

B REG - INF4 - DDFB8X

C REG - INF3 - DDFB8W

X REG - INF2 - DDFB8M

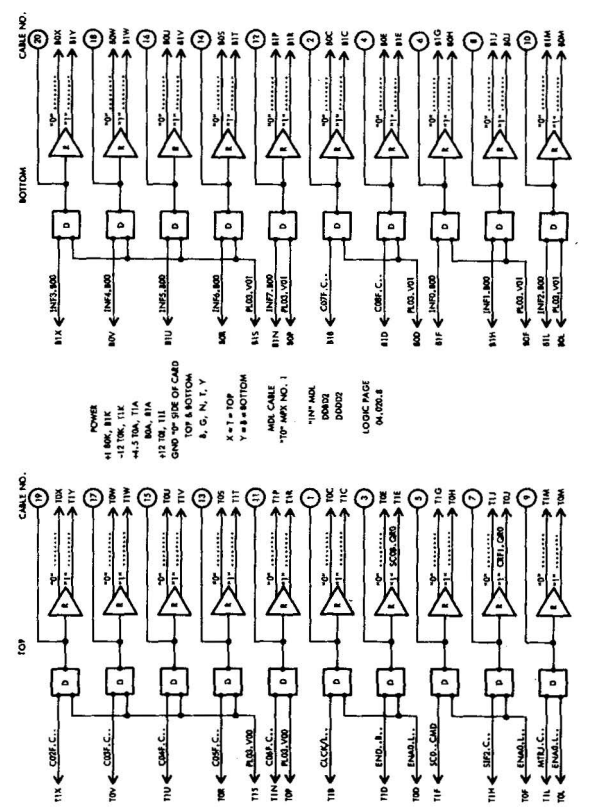
A REG - INF5 - DDFB9W

Y REG - INF1 - DDFB9N

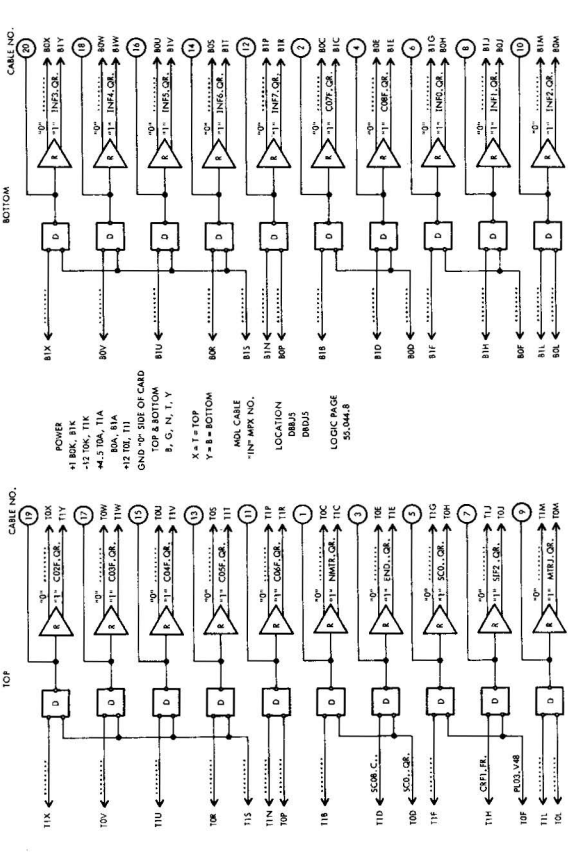
(See Note on page 8-45.)

DISPLAY AND MDL

MDL to Multiplexor No. 1 Interface

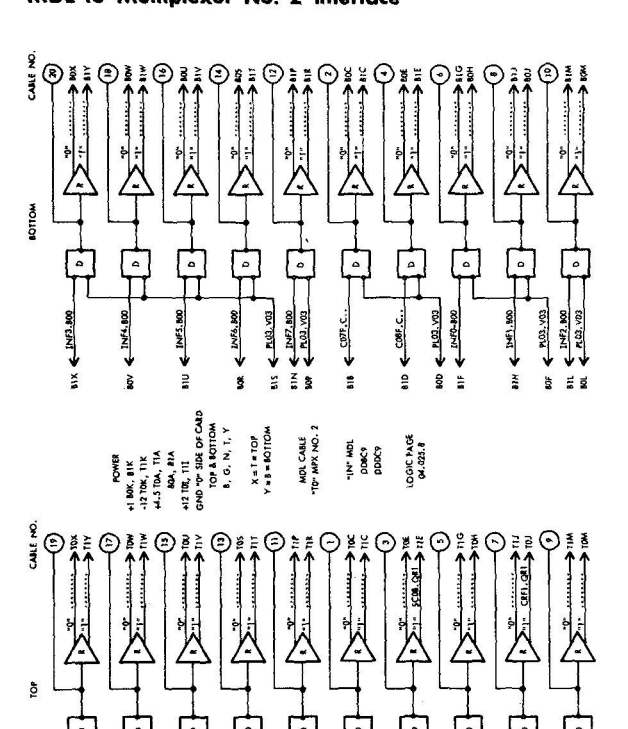


Multiplexor No. 1 to MDL Interface

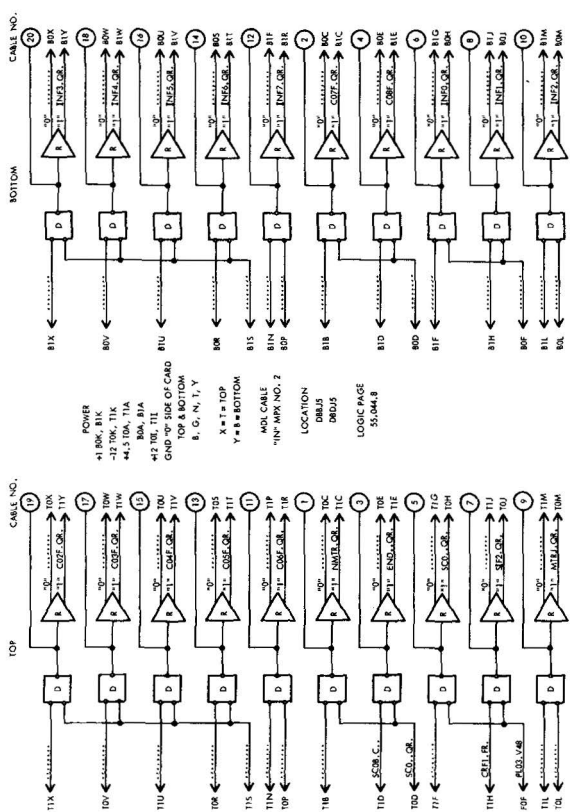


DISPLAY AND MDL

MDL to Multiplexor No. 2 Interface

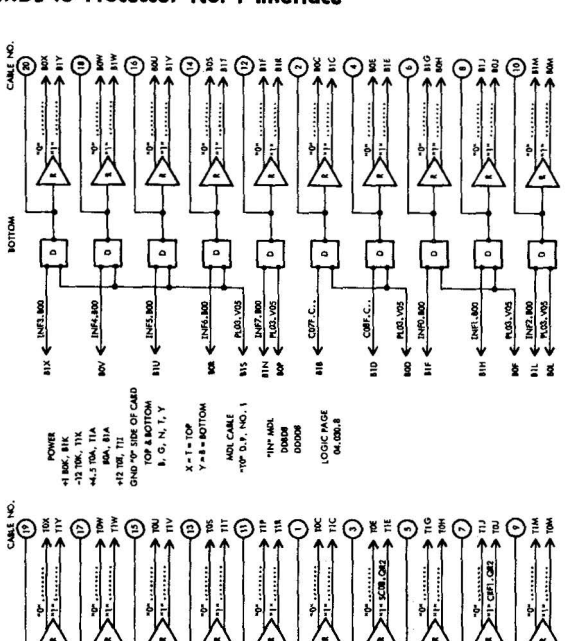


Multiplexor No. 2 to MDL Interface

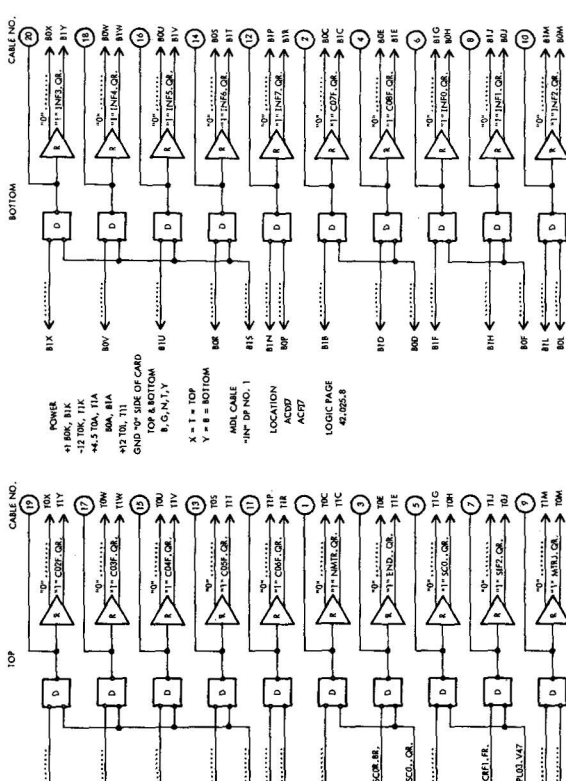


DISPLAY AND MDL

MDL to Processor No. 1 Interface

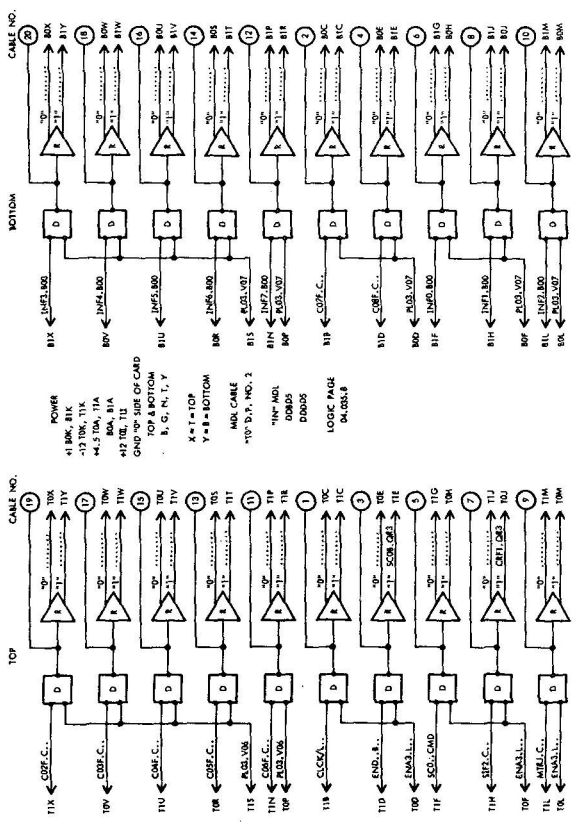


Processor No. 1 to MDL Interface

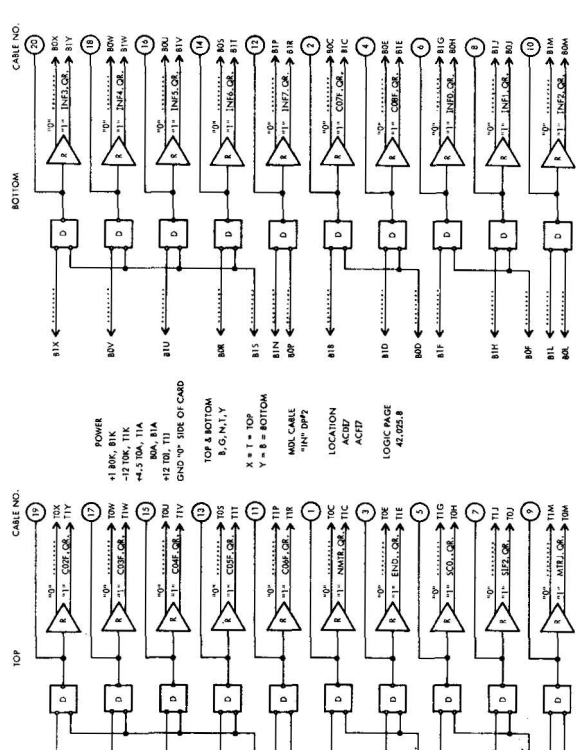


DISPLAY AND MDL

MDL to Processor No. 2 Interface



Processor No. 2 to MDL Interface



CONSOLE DISPLAY CONTROL

NOMINAL MULTI OUTPUTS

NOTE

All multi adjustments are on clock timer card in location BABF2.

Multi	Nominal Output	Point Observed	Control
CLKL	833 usec between pulses (1200 baud) or 416 usec between pulses (2400 baud)	BABF2U	Bottom
DLTO	1.5 sec pulse width (positive portion)	BABF2, chip C0, pin H	Middle
CHTO	1.5 sec pulse width (positive portion)	BABF2, chip A0, pin H	Top

If adjustment is required, refer to Section IV of FETM.

TRANSLATOR OUTPUT TEST POINTS

ASCII-to-EBCDIC

Level	Test Point
X01E	BABK4M
X02E	K5C
X03E	K7X
X04E	K6W
X05E	K9C
X06E	K9D
X07E	K4B
X08E	K8X

EBCDIC-to-ASCII

Level	Test Point
XEA1	BABJ8D
XEA2	J8E
XEA3	J8C
XEA4	K0E
XEA5	K1F
XEA6	K2B
XEA7	K2E

CARD LOADING

Location	Name
BABA1	LEVEL CHANGER/EXCHANGE 00
BABA4	LEVEL CHANGER/EXCHANGE 01
BABA7	LEVEL CHANGER/EXCHANGE 02
BABB0	LEVEL CHANGER/EXCHANGE 03
BABB3	LEVEL CHANGER/EXCHANGE 04
BABB6	LEVEL CHANGER/EXCHANGE 05
BABB9	LEVEL CHANGER/EXCHANGE 06
BABC2	LEVEL CHANGER/EXCHANGE 07
BABC5	RDY STATUS CABLE DRIVER
BABC9	INPUT REQ COAX DRIVER

CARD LOADING (Cont)

Location	Name
BABF2	SINGLE LINE TIMERS
BABF4	VOLTAGE DISTRIBUTION
BABF6	LINE CONTROL F/F'S NO. 2
BABF8	LINE CONTROL F/F'S NO. 1
BABG0	SHIFT REGISTER, UPPER
BABG2	SHIFT REGISTER, LOWER
BABG6	MTR/DISPLAY NO. 4
BABH0	B/C REGISTER
BABH2	CHARACTER AND PARITY GENERATOR
BABH4	CHARACTER SENSE
BABH8	S/R CONTROL AND MISC.
BABI2	MTR/DISPLAY NO. 3
BABI6	CONTROL STATE REGISTER
BABI8	C/S REGISTER DECODER
BABJ0	C/S REGISTER CONTROL NO. 1
BABJ2	C/S REGISTER CONTROL NO. 2
BABJ6	ERROR CONTROL
BABJ8	TRANSLATOR (E to A) NO. 1*
BABK0	TRANSLATOR (E to A) NO. 2*
BABK2	TRANSLATOR (E to A) NO. 3*
BABK4	TRANSLATOR (A to E) NO. 1*
BABK6	TRANSLATOR (A to E) NO. 2*
BABK8	TRANSLATOR (A to E) NO. 3*
BABM8	CLOCK AND LOCAL MAINT.
BABN4	ERROR REGISTER
BABN8	MTR/DISPLAY NO. 2
BABP2	SEQ CTR REGISTER
BABP4	U/D CONFIGURATOR CARD
BABP6	U/D CONFIGURATOR CARD
BABP8	U/D AND OP. REGISTERS
BABQ2	MTR/DISPLAY NO. 1
BABQ6	INTERFACE CONTROL
BABQ8	INFO REGISTER IN "C"
BABR0	INFO REGISTER IN "B"
BABR2	INFO REGISTER IN "A"
BABR6	INFJ
BABR8	INFJ

* E = Console Display Control; A = Terminal Unit.

VOLTAGE TEST POINTS

Voltage	Test Point
+4.75V	Any 0A/1A Pin
-2.0	Any 0Z/1Z Pin
+12	BABF4K, F4L, F4M, F4N
-12	BABF4C, F4D, F4E, F4F

CONSOLE DISPLAY CONTROL

SWITCH/INDICATOR PLUG-ONS

BABM8	BABQ2		BABN8	BAB12		BABG6	
LOCAL	IN A	A8	ARF	SHIFT REG	M	EXTF	
REMOTE		A4	AGF		8	B C R	7
SINGLE PLSE		A2	TMOF		7		6
SINGLE INST		A1	DPEF		6		5
RUN	IN B	B8	COCF		5		4
START		B4	NRYF		4		3
		B2	ROVF		3		2
CLEAR		B1	MAEF		2		1
IOCB OFF CDL	CLEAR		CLEAR	CLEAR		CLEAR	
	IN C	C8	MC1F	S R	1	XMTF	
STCB OFF		C4	MC2F		S	CHTF	
		C2	OPR 2			SYNF	
REQ TO SEND		C1	1	C S R	16	ITLF	
	UDR	U8	SCR 8		8	CTSF	
NOT USED		U4	4		4	STXF	
NOT USED		U2	2		2	UPAF	
		U1	1		1	CHRF	
	CLEAR		CLEAR	CLEAR		CLEAR	

SWITCH FUNCTIONS

Name	Function
LOCAL REMOTE	In LOCAL, disables control and information levels between MPX and Console Display Control.
SINGLE PLSE SINGLE INST RUN	Controls single-pulse and run modes; single-instruction mode is not implemented.
START	Initiates single-pulse or run mode.
CLEAR	Clears all indicators in unit.
IOCB OFF CDL	In IOCB, causes true level on I/O Complete bus; in CDL, allows Channel Designate level for control data transfer.
STCB OFF	Allows Start Channel Bus level to be true.
REQ TO SEND	Enables Request To Send level.

DATA COMMUNICATIONS SYSTEM

DCP VOLTAGE TEST POINTS

-12 Volt Circuits

M12V. R1	FB8F5C	FBDH1R
M12V. R2	5D	DH1S
NG12.V02	5E	BE3K
NG12.V03	5F	BG1K
NG12.V04	5G	BE6K
NG12.V05	5H	BG7K
NG12.V06	51	BF3K
NG12.V07	5J	
NG12.V08	5K	BF2K
NG12.V09	5L	
NG12.V10	5M	BG8K
NG12.V11	5N	BE9K
NG12.V12	5P	BF0K
NG12.V13	5Q	BG4K
NG12.V14	5R	BG5K
NG12.V15	5S	BE7K
NG12.V16	5T	BG2K
NG12.V17	5U	BE4K
NG12.V18	5V	BF8K
NG12.V19	5W	BF9K

+1 Volt Circuits

PL01.V00	FBDF6C	
PL01.V01	6D	
PL01.V02	6E	FBDE3K
PL01.V03	6F	DG1K
PL01.V04	6G	DE6K
PL01.V05	6H	DG7K
PL01.V06	61	DF3K
PL01.V07	6J	
PL01.V08	6K	DF2K
PL01.V09	6L	
PL01.V10	6M	DG8K
PL01.V11	6N	DE9K
PL01.V12	6P	DF0K
PL01.V13	6Q	DG4K
PL01.V14	6R	DG5K
PL01.V15	6S	DE7K
PL01.V16	6T	DG2K
PL01.V17	6U	DE4K
PL01.V18	6V	DF8K
PL01.V19	6W	DF9K

DATA COMMUNICATIONS SYSTEM

DCP VOLTAGE TEST POINTS (Cont)

+12 Volt Circuits

PL12.V00	FBBF6C	FBBF8I
PL12.V01	6D	BF9I
PL12.V02	6E	DH1F
PL12.V03	6F	DH1G
PL12.V04	6G	BE9I
PL12.V05	6H	FD2K
PL12.V06	6I	BF2I
PL12.V07	6J	FD2L
PL12.V08	6K	BF3I
PL12.V09	6L	BG7I
PL12.V10	6M	BG8I
PL12.V11	6N	BF0I
PL12.V12	6P	BG4I
PL12.V13	6Q	BG5I
PL12.V14	6R	BE6I
PL12.V15	6S	BE7I
PL12.V16	6T	BG1I
PL12.V17	6U	BG2I
PL12.V18	6V	BE3I
PL12.V19	6W	BE4I

+3 Volt Circuits

PL03.V01	FBDF5D	FBDE3D
PL03.V09	5L	DE3L

-4.5 Volt Circuits

M4.5.V00	FBDF5X	FBDG7I
M4.5.V01	5Y	DG8I

DCP CLUSTER WRITE CLOCK STROBE TEST POINTS

Name	End of Net Test Point
WSTB.L01	FALA0L
WSTB.L02	FALB2L
WSTB.L03	FAJB2L
WSTB.L04	FAJA0L
WSTB.L05	FAJC0G
WSTB.L06	FAJC1H

DCP CLUSTER CLOCK TEST POINTS

NOTE

Refer to LDAS Logic Book page 80.150.8
for additional details.

Name	End of Net Test Point
CLK1.CLK	FABB0B
CLK2.CLK	FABB6B
CLK3.CLK	FABC6B
CLK4.CLK	FADA6B
CLK5.CLK	FADC6B
CLK6.CLK	FADC2B
CLK7.CLK	FAFC6B
CLK8.CLK	FAHC6B
CL11.CLK	FAJC6B
CL12.CLK	FALC6B

(200ns between leading edges; 25ns ±3ns pulse width at +1 volt level

DCP CLOCK TEST POINTS

Name	End of Net Test Point
CP01....	FBHA0B
CP02....	FBHB4B
CP03....	FBHD6B
CP04....	FBJF4B
CP05....	FBFF1B
CP06....	FBFE1B
CP07....	FBFF3B
CP08....	FBFH0B
CP09....	FBDA0B
CP10....	FBDE0B
CP11....	FBHC6B
CP12....	FBHA6B

Clock Receiver location: FBDH0
Clock Backplane Driver location: FBFD1

DATA COMMUNICATIONS SYSTEM

DCP/CLUSTER INTERFACE TEST POINTS

GROUND	1	A	GROUND
CAN-0	2	B	DES-0
CAN-1	3	C	DES-1
CAN-2	4	D	DES-2
CAN-3	5	E	DES-3
Key: no lead	6	F	Key: no lead
ACG	7	H	HLD
CWR	8	J	IWR
AA3	9	K	AA2
AA1	10	L	AA0
GROUND	11	M	GROUND
(Spare)	12	N	(Spare)
(Spare)	13	P	AC4
AC3	14	R	AC2
AC1	15	S	AC0
AI7	16	T	AI6
AI5	17	U	AI4
AI3	18	V	AI2
AI1	19	W	AI0
AI8	20	X	(Spare)
(Spare)	21	Y	CLR
GROUND	22	Z	GROUND

DCP CLUSTER TEST POINTS

Level	Test Point	Level	Test Point	Level	Test Point
RT = 00	FABA1Q	TY0F	FAFC3G	C100F	FADA7U
RT = 01	FABA0Q	TY1F	FAFC2E	C101F	FADA7R
RT = 02	FABA1L	TY2F	FAFC2W	C102F	FADA7H
RT = 03	FABA0L	TY3F	FAFC3F	C103F	FADA7X
RT = 04	FABA0I	TY4F	FAFC3D	C104F	FADA7N
RT = 05	FABA1I			C105F	FADA6U
RT = 06	FABA1F	CT0F	FADB9V	C106F	FADA6J
RT = 07	FABA1G	CT1F	FADB8V	C107F	FADA7E
RT = 08	FABA0X			C108F	FADA6H
RT = 09	FABA0Y	BT0F	FADA0N	C109F	FADA6K
RT = 10	FABA0U	BT1F	FADA1P	C110F	FADA6M
RT = 11	FABA0V	BT2F	FADA1G		
RT = 12	FABA1R	BT3F	FADA0F	C2QF	FADB4F
RT = 13	FABA0F	BT4F	FADA0H	C20F	FADB4N
RT = 14	FABA0H	BT5F	FADA1C	C21F	FADB4T
RT = 15	FABA0G	BT6F	FADA0C	C22F	FADB4W
RT0F	FABA1T			C23F	FADB5E
RT1F	FABA1S	PT0F	FAFB7X	C24F	FADB4E
RT2F	FABA0R	PT1F	FAFB6W	C25F	FADB4U
RT3F	FABA0S	PT2F	FAFB7W	C26F	FADB5R
		PT3F	FAFB6Y	C27F	FADB5G
BC0F	FAHA3Q	PT4F	FAFB6X	C28F	FADB4D
BC1F	FAHA2C	PT5F	FAFB6E		
BC2F	FAHA3T	PT6F	FAFB6M	FIPF	FAFC3J
		PT7F	FAFB6R	SSCF	FADC0D
B10F	FAHC6Y			SSTF	FADC0Y
B11F	FAHC6W	SC0F	FAHB1S	ACQF	FAFC5X
B12F	FAHC6T	SC1F	FAHB1T	ACPF	FAFB8W
B13F	FAHC6D	SA0F	FAHB1I		
		SA1F	FAHB0X		
		SA2F	FAHB1M		

DATA COMMUNICATIONS SYSTEM

DCP LOCAL MEMORY POWER SUPPLY INTERFACE

Pin	Function
TB1-5	+5V (Memory
TB1-6	+5VR
TB1-1	+15V
TB1-2	+15VR
TB1-3	+15V
TB1-4	+15VR
TB1-7	+5V (Interface)
TB1-8	+5VR
TB2-1	-16V
TB2-2	-16VR
TB2-7	$\overline{DS}$
TB2-8	$\overline{DSR}$
TB2-5	Input +15V for X-Y*
TB2-6	Output +15V for X-Y*
TB2-3	TH
TB2-4	THR

* Normally connected by jumper

DCP LOCAL MEMORY MODULE SELECT JUMPERS

Jumper		Module	Jumpers on I/O Card
LAI3	LAI2		
0	0	0	3-5, 1-6
0	1	1	3-5, 2-6
1	0	2	4-5, 1-6
1	1	3	4-5, 2-6

DCP ADAPTER TEST POINTS

	05F	04F	03F	02F	01F	00F	19F	18F	17F	16F	15F	14F	13F	12F	11F	10F	
	0J	0I	0E	1E	0D	0B	-	-	0V	0U	1T	0S	0R	1Q	0N	0M	Interface TP
	1J	1I	1G	1F	1D	0C	1K	1M	0X	0W	0T	1S	1R	0Q	0P	1P	Logic Pin
Type I RS232	CD	CA	ED	SA	RS	BA	CD	CA	SB	DD	DB	CF	CE	CC	CB	BB	Name
	$D	$L	$M	$C	#H	$K			$P	$V	$X	$J	$H	$Y	$R	$W	Conn. Pin
	20	4	24	11	21	2			12	17	15	8	22	6	5	3	Data Set Conn.
Type II 2 Wire						BA					TW					BB	Name
						$S										$W	Conn. Pins
						$E										$E	
						1										1	Data Set Conn.
Type III ACU	DPR	CRQ	NBB	NB4	NB2	NB1	DPR	CRQ	PND	-	-	-	DLO	PWI	COS	ACR	Name
	$K	$L	$V	$T	$X	$Z			$R				$H	$Y	$S	$W	Conn. Pin
	2	4	17	16	15	14			5				22	6	13	3	Data Set Conn.
Type IV TT	CD	DR	ATT	QQS	ABA	ABB	CD	DR	D4	D3	-	CF	CE	CC	D2	D1	Name
	$H	#H	$X	#S	$D	$E			$Y	$R		$T	$Z	#P	$L	$W	Conn. Pin
	22	21	15	25	20	19			6	5		16	14	23	4	3	Data Set Conn.
Type V VR	RS	SP	-	-	-	D1	-	-	-	WP	-	PP	-	RY	-	-	Name
	$S	$L				$K				$V		$J		$Y			Conn. Pin
	Designate OK (SK00.BR1)									Designate +2				IH (SD02.BR1)			

NOTE: Answer Back leads on Touch Tone and Voice Response:

				VR
VAB2.L00 ——— VRA	1V	$V ——— 1V		#Z
VAB2.L00 ——— VRB	1U	$U ——— 1U		#Y

DATA COMMUNICATIONS SYSTEM

DCP LOCAL MEMORY ASSEMBLY LOCATIONS

Location	Assembly	Use
J1-3	Not Used	
J4	Time Delay Multi (TDM)	Control
J5	Memory Timing and Control (MTC)	Control
J6	Current Source Module (CSM)	X and Y Current Source +12V Threshold
J7	Address Drive Switch (ADS)	Address Bits 0, 1, 2
J8	Address Drive Switch (ADS)	Address Bits 3, 4, 5
J9	Address Drive Switch (ADS)	Address Bits 6, 7, 8
J10	Address Drive Switch (ADS)	Address Bits 9, 10, 11
J11	Not Used	
J12-18	Stack	4K Words X 52 Bits
J19	Not Used	
J20	Data Circuit Module (DCM)	Data Bits 0-3
J21	Data Circuit Module (DCM)	Data Bits 4-7
J22	Data Circuit Module (DCM)	Data Bits 8-11
J23	Data Circuit Module (DCM)	Data Bits 12-15
J24	Data Circuit Module (DCM)	Data Bits 16-19
J25	Data Circuit Module (DCM)	Data Bits 20-23
J26	Data Circuit Module (DCM)	Data Bits 24-27
J27	Data Circuit Module (DCM)	Data Bits 28-31
J28	Data Circuit Module (DCM)	Data Bits 32-35
J29	Data Circuit Module (DCM)	Data Bits 36-39
J30	Data Circuit Module (DCM)	Data Bits 40-43
J31	Data Circuit Module (DCM)	Data Bits 44-47
J32	Data Circuit Module (DCM)	Data Bits 48-51

DCP LOCAL MEMORY ASSEMBLY LOCATIONS (Cont)

Location	Assembly	Use
J33	I/O Control Module	Addresses, Misc. Signals In and Out
J34	I/O Data Module	Data Interfacing and Line Drivers
J35	I/O Data Module	Data Interfacing and Line Drivers

DCP LOCAL MEMORY INTERFACE SIGNAL CONNECTIONS

Signal Connector		Signal Connector Return		Signal Designation
J33	$K	J33	$A	LA00
	$J			LA01
	$E			LA02
	#D			LA03
	#C			LA04
	#E			LA05
	$U			LA06
	$T			LA07
	$V			LA08
	#U			LA09
	$P			LA10
	$N			LA11
	$R			LA12
	#B			LA13
	#J			LREQ
	#P			LMRD
	#H			LMWT
	$H			LMA0
	#L			LMER
	$C			LMPR
J33	#K	J33	#Z	LMBZ

Signal Connector		Signal Connector Return		Signal Designation
J34	#H	J34	$A	LM00
	$D			LM01
	#D			LM02
	#C			LM03
	$B			LM04
	#B			LM05
	$H			LM06
	$M			LM07
	#E			LM08
	$C			LM09
	$E			LM10
	$K			LM11
	$S			LM12
	$W			LM13
	#X			LM14
	#T			LM15
	$T			LM16
	$V			LM17
	$X			LM18
	$P			LM19
	$Y			LM20
	#Y			LM21
	#W			LM22
	#U			LM23
	$U			LM24
J34	#V	J34	#Z	LM25

DATA COMMUNICATIONS SYSTEM

DCP LOCAL MEMORY INTERFACE SIGNAL CONNECTIONS (Cont)

Signal Connector	Signal Connector Return	Signal Designation	Signal Connector	Signal Connector Return	Signal Designation
J35 #H	J35 $A	LM26	J35 $W	J35 $A	LM39
$D		LM27	#X		LM40
#D		LM28	#T		LM41
#C		LM29	$T		LM42
$B		LM30	$V		LM43
#B		LM31	$X		LM44
$H		LM32	$P		LM45
$M		LM33	$Y		LM46
#E		LM34	#Y		LM47
$C		LM35	#W		LM48
$E		LM36	#U		LM49
$K		LM37	$U		LM50
J35 $S	J35 #Z	LM38	J35 #V	J35 #Z	LM51

DCP LOCAL MEMORY TIMING AND CONTROL TEST POINTS

Test Point	Function or Circuit Designation
TP 1	Timing Pulse 1 (Delay Line In)
TP 2	Y Read Timing Pulse Bar ($\overline{YRT}$)
TP 3	Cycle Initiate FF
TP 4	Clear Write (CW)
TP 5	Post Read Phase (PRϕ)
TP 7	Read Phase (Rϕ)
TP 8	Write Phase (Wϕ)
TP 9	Memory Busy Bar ($\overline{MB}$)
TP 10	MA0
TP 11	Reset Data Register Bar ($\overline{RDR}$)
TP 12	Not Used
TP 14	Y Write Timing Pulse Bar ($\overline{YWT}$)
TP 15	Inhibit (INH)
TP 16	General Reset Bar ($\overline{GR}$)
TP 17	Ground

DCP LOCAL MEMORY CURRENT SOURCE TEST POINTS

Test Point	Level or Pulse
TP 1	Y Grounding Switch
TP 2	+15V
TP 3	Y Regulating Switch
TP 4	Ground
TP 5	X Grounding Switch
TP 6	+12V
TP 7	X Regulating Switch

DCP LOCAL MEMORY ADDRESS BITS TEST POINTS

Test Point	Address Drive Switch			
	J7	J8	J9	J10
	Address Bits			
7	0	3	6	9
4	1	4	7	10
9	2	5	8	11
Decode Designation	XB	XA	YC	YD

DATA COMMUNICATIONS SYSTEM

DCP LOCAL MEMORY DATA BITS TEST POINTS

Circuit Designation	Signal Designation	Test Point	J20	J21	J22	J23	J24	J25	J26	J27	J28	J29	J30	J31	J32
			Data Bits												
A	SA OUT	1	0	4	8	12	16	20	24	28	32	36	40	44	48
	DR OUT	2													
B	SA OUT	3	1	5	9	13	17	21	25	29	33	37	41	45	49
	DR OUT	4													
C	SA OUT	6	2	6	10	14	18	22	26	30	34	38	42	46	50
	DR OUT	7													
D	SA OUT	9	3	7	11	15	19	23	27	31	35	39	43	47	51
	DR OUT	10													

SA = Sense Amp

DR = Data Register

DCP PROGRAM REGISTER TEST POINTS

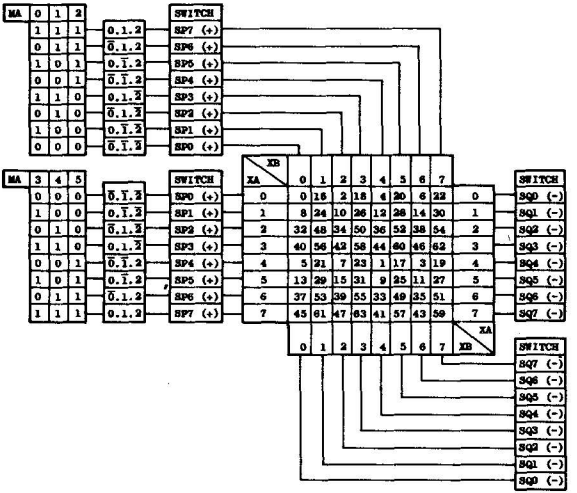
P-R		P-L		TAGS	
P00	FBHA7L	P24	FBHA3T	P48	FBHA1L
P01	HA7M	P25	HA2E	P49	HA1M
P02	HA7G	P26	HA9T	P50	HA1G
P03	HA7S	P27	HA8E	P51	HA1S
P04	HB7L	P28	HB3T		
P05	HB7M	P29	HB2E		
P06	HB7G	P30	HB9T		
P07	HB7S	P31	HB8E		
P08	HC7L	P32	HC3T		
P09	HC7M	P33	HC2E		
P10	HC7G	P34	HC9T		
P11	HC7S	P35	HC8E		

DCP PROGRAM REGISTER TEST POINTS (Cont)

P-R		P-L		TAGS
P12	HD7L	P36	HD3T	
P13	HD7M	P37	HD2E	
P14	HD7G	P38	HD9T	
P15	HD7S	P39	HD8E	
P16	FD8L	P40	FD4T	
P17	FD8M	P41	FD3E	
P18	FD8G	P42	FE0T	
P19	FD8S	P43	FD9E	
P20	FE8L	P44	FE4T	
P21	FE8M	P45	FE3E	
P22	FE8G	P46	FF0T	
P23	FE8S	P47	FE9E	

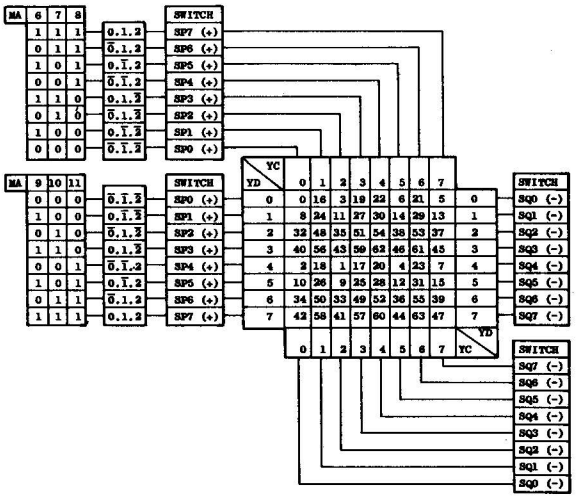
DATA COMMUNICATIONS SYSTEM

DCP MEMORY X DECODE: OCTAL TO LINE CONVERSION



XA \ XB	0	1	2	3	4	5	6	7	
0	0	16	2	18	4	20	6	22	0
1	8	24	10	26	12	28	14	30	1
2	32	48	34	50	36	52	38	54	2
3	40	56	42	58	44	60	46	62	3
4	5	21	7	23	1	17	3	19	4
5	13	29	15	31	9	25	11	27	5
6	37	53	39	55	33	49	35	51	6
7	45	61	47	63	41	57	43	59	7
XA \ XB	0	1	2	3	4	5	6	7	

DCP MEMORY Y DECODE: OCTAL TO LINE CONVERSION



DATA COMMUNICATIONS SYSTEM

DCP CLUSTER MASK FLIP-FLOP TEST POINTS

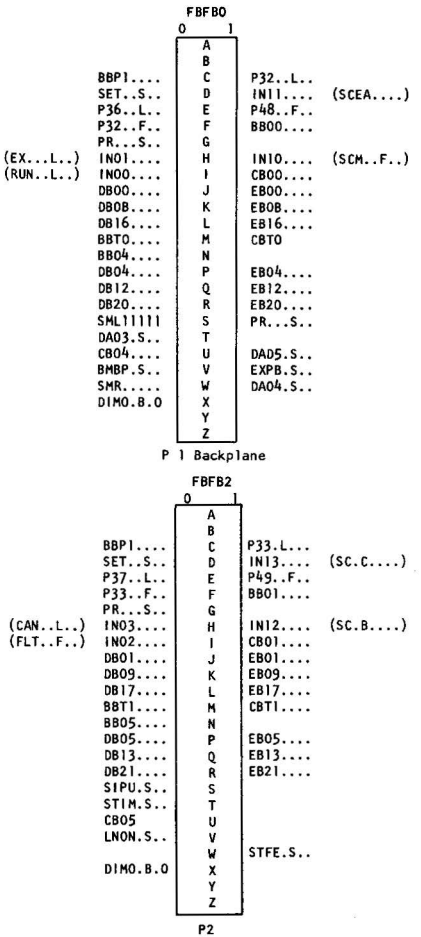
FF	Output Terminal	Location
CM00	EXEN..0A	BA0C
01	EXEN..1A	BA0M
02	EXEN..2A	BA0N
03	EXEN..3A	BA0I
04	EXEN..0B	DA1C
05	EXEN..1B	DA0J
06	EXEN..2B	DA1F
07	EXEN..3B	DA0H
08	EXEN..0C	DA0K
09	EXEN..1C	DA1I
10	EXEN..2C	DA1J
11	EXEN..3C	DA0M
12	EXEN..0D	BA0F
13	EXEN..1D	BA1F
14	EXEN..2D	BA1I
15	EXEN..3D	BA1Q

DCP ADDRESS REGISTER TEST POINTS

B 6700 Address	Source
BA00	DF9X
BA01	BF8V
BA02	BF9U
BA03	BF8R
BA04	BF9N
BA05	BF9B
BA06	BF9D
BA07	BF9F
BA08	BF9H

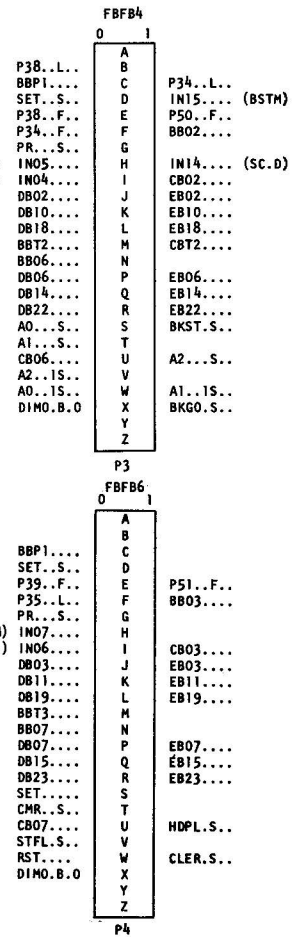
B 6700 Address	Source
BA09	BF9L
BA10	BF9X
BA11	DF8V
BA12	DF9U
BA13	DF8R
BA14	DF9N
BA15	DF9B
BA16	DF9D
BA17	DF9F
BA18	DF9H
BA19	DF9L

DCP DISPLAY INTERFACE TEST POINTS

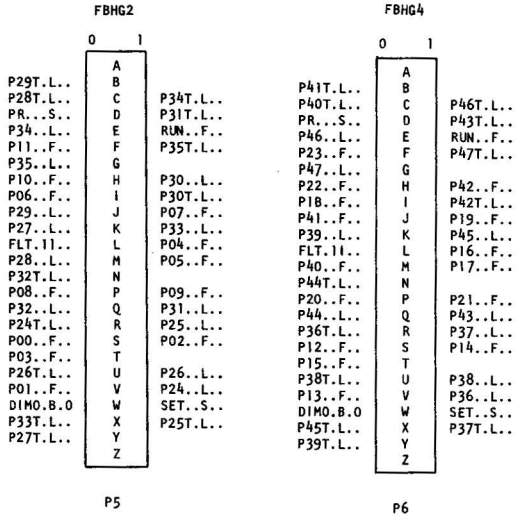


DATA COMMUNICATIONS SYSTEM

DCP DISPLAY INTERFACE TEST POINTS (Cont)

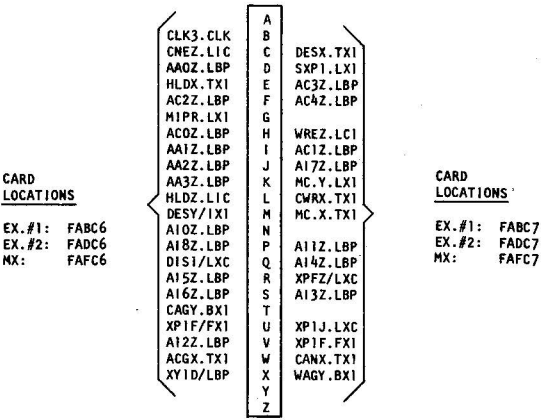


DCP DISPLAY INTERFACE TEST POINTS (Cont)



DATA COMMUNICATIONS SYSTEM

DCP CLUSTER EXCHANGE TEST POINTS



DCP CLUSTER DISCRETE VOLTAGE TEST POINTS

-12 Volt Circuits

Name	Source	Destination
MI12.V01		
MI12.V02		
MI12.V03	HA0E	JA1W
MI12.V04	HA0F	JA4W
MI12.V05	HA0G	JA7W
MI12.V06	HA0H	JB0W
MI12.V07	HA0I	JB3W
MI12.V08	HA0J	JB6W
MI12.V09	HA0K	JB9W
MI12.V10	HA0L	LA1W
MI12.V11	HA0M	LA4W
MI12.V12	HA0N	LA7W
MI12.V13	HA0P	LB0W
MI12.V14	HA0Q	LB3W
MI12.V15	HA0R	LB6W
MI12.V16	HA0S	LB9W
MI12.V17	HA0T	LC2W
MI12.V18	HA0U	LC5W
MI12.V19	HA0V	LC7W

+12 Volt Circuits

Name	Source	Destination
PL12.V01	HA1C	FA1K
PL12.V02	HA1D	FA1L
PL12.V03	HA1E	JA1X
PL12.V04	HA1F	JA4X
PL12.V05	HA1G	JA7X
PL12.V06	HA1H	JB0X
PL12.V07	HA1I	JB3X
PL12.V08	HA1J	JB6X
PL12.V09	HA1K	JB9X
PL12.V10	HA1L	LA1X
PL12.V11	HA1M	LA4X
PL12.V12	HA1N	LA7X
PL12.V13	HA1P	LB0X
PL12.V14	HA1Q	LB3X
PL12.V15	HA1R	LB6X
PL12.V16	HA1S	LB9X
PL12.V17	HA1T	LC2X
PL12.V18	HA1U	LC5X
PL12.V19	HA1V	LC7X

DATA COMMUNICATIONS SYSTEM

ADAPTER CLUSTER INPUT REGISTER BIT ASSIGNMENTS

	ASYNCHRONOUS	SYNCHRONOUS	AUTO CALL	TOUCH TONE	VOICE RESPONSE
IR9F	CD- DATA TERM READY	CD- DATA TERM READY	DPR- DIGIT PRESENT	CD- DATA TERM READY	
IR8F	CA- REQUEST TO SEND	CA- REQUEST TO SEND	CRQ- CALL REQUEST	DR- DATA RECEIVE	
IR7F	SB- SUPERVISORY REC. DATA	SB- SUPERVISORY REC. DATA	PND- PRESENT NEXT DIGIT	D4- DATA BIT 4	
IR6F	CZ- RESTRAINT RECEIVED	DD- RECEIVE CLOCK		D3- DATA BIT 3	WP- WORD PULSE
IR5F	TW- TWO WIRE	DB- TRANSMIT CLOCK			
IR4F	CF- CARRIER DETECT	CF- CARRIER DETECT		CF- CARRIER	PP- PHRASE PULSE
IR3F	CE- RING INDICATOR	CE- RING INDICATOR	DLO- DATA LINE OCCUPIED	CE- RING INDICATOR	
IR2F	CC- DATA SET READY	CC- DATA SET READY	PWI- POWER INDICATION	CC- DATA SET READY	RY- READY
IR1F	CB- CLEAR TO SEND	CB- CLEAR TO SEND	COS- CALL ORIG. STATUS	D2- DATA BIT 2	
IR0F	BB- DATA RECEIVE	BB- DATA RECEIVE	ACR- ABANDON CALL/RETRY	D1- DATA BIT 1	

ADAPTER CLUSTER OUTPUT REGISTER BIT ASSIGNMENTS

	ASYNCHRONOUS	SYNCHRONOUS	AUTO CALL	TOUCH TONE	VOICE RESPONSE
OR5F	CD- DATA TERM READY	CD- DATA TERM READY	DPR- DIGIT PRESENT	CD- DATA TERM READY	RS- RESET
OR4F	CA- REQUEST TO SEND	CA- REQUEST TO SEND	CRQ- CALL REQUEST	DR- DATA RECEIVE	SP- SHIFT PULSE
OR3F	ED- FAST DISCONNECT		NBB- NUMBER BIT 8	ATT- ATTENDANT	
OR2F	SA- SUPERVISORY TRANS. DATA	SA- SUPERVISORY TRANS. DATA	NB4- NUMBER BIT 4	OOS- OUT OF SERVICE	
OR1F	RS- RESTRAINT TRANSMIT		NB2- NUMBER BIT 2	ABA- ANSWER BACK A	
OR0F	BA- DATA TRANSMIT	BA- DATA TRANSMIT	NB1- NUMBER BIT 1	ABB- ANSWER BACK B	DI- DATA INPUT

ADAPTER CLUSTER MAINTENANCE REGISTER BIT ASSIGNMENTS

Bit	Function
MR6F	Disconnect Exchange No. 1 } See Note 1
MR5F	Disconnect Exchange No. 2 }
MR4F	Inhibit IR bus input to IRnF
MR3F,2F = 3	Return next Write information to DCP (see Note 2)
= 2	Display CIR on maintenance exchange (see Note 3)
= 1	Inhibit writing in IC memory
= 0	Idle
MR1F,0F = 3	Inhibit positional interrupt priority
2	Open maintenance exchange window (see Note 4)
1	Set Single Step Translation flip-flop to 1
0	Idle

NOTE

1. When either exchange is disconnected, CAN's to the associated DCP are inhibited. An exchange can logically be disconnected only if the maintenance exchange is connected.

2. When MR3F, 2F equal 3, the next write information from a DCP will be returned to the DCP with a CAN. MR3F, 2F are reset to 0.

3. Must be used in conjunction with Maintenance Display panel.

4. Maintenance Exchange crosspoint is normally set every 12.6 microseconds and only during a 200 nanosecond window. When MR1F, 0F equal 2, the timing restriction is removed and the maintenance exchange can access with only conflict restrictions.

DATA COMMUNICATIONS SYSTEM

TIMER ADAPTER TIMING PERIOD

Toggle Bits	Time Period (MS)	Normal Multiplier Range in MS(1-31)	Expanded Multiplier Range in MS(32-62)
000	0.0064	0.0064 - 0.192	13.11 - 25.40
001	0.0512	0.192 - 1.536	52.43 - 101.6
010	0.4096	1.536 - 12.228	209.7 - 406.3
011	3.27	12.228 - 98.1	838.9 - 1625
100	26.2	98.1 - 786	3355 - 6501
101	209.7	786 - 6291	13422 - 26005
110	1677	6291 - 50310	53687 - 104019
111	13421	50310 - 402630	214748 - 416075

MAINTENANCE DISPLAY CLUSTER ADDRESSING

Adapter Control Register [2:3]	Field	Adapter Information Register (Length)	Miscellaneous
0	Type	[4:5]	[5:1]=F.I.P.F.
1	C2	[8:9]	CC3F = C2QF
2	BC/BI	[6:3] [3:4]	CC3F=Clean House
3	SC/SA	[4:2] [2:3]	COOF=[5:1]; C1OF=[6:1]
4	CT/BT	[8:2] [6:7]	
5	C1	[8:9]	
6	IR	[8:9]	CC3F = IR9F
7	MR	[6:7]	
3*AC3F	PT	[7:8]	Write only in PT field
7*AC3F	OR	[7:2] [4:2] [1:2]	Interrogate OR only

CTμL INTEGRATED CIRCUITS

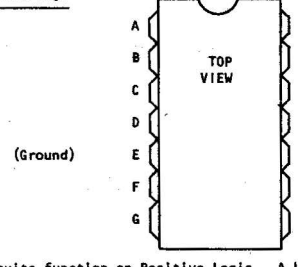
There are six basic CTμL logic circuit elements: The "And" gate, JK Flip-Flop, RS Flip-Flop, Memory Cell, Inverter and Buffer. These circuit elements are employed in 11 unique packages. These packages (usually called IC's or chips) have 14 pins and can have from 1 to 4 independent circuits built on the same silicon chip. The CTμL circuits (Complementary Transistor Micro Logic) consist of PNP and NPN transistors and resistors integrated on a silicon chip.

The chip is housed in a ceramic package which has a key at one end. The part number of each chip type is stamped on the top of the package.

The voltages required for CTμL IC's are +4.5, -2 and ground. The voltage pins are the same for all IC packages, except the memory cell which does not use -2V.

+4.5V	Pin M
-2V	Pin L
Ground	Pin E

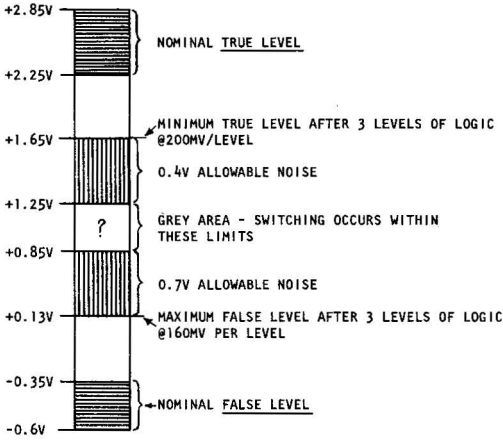
IC Pin Numbering:



CTμL circuits function on Positive Logic. A "True" level is approximately +2.5V, a "False" level is around -.5V. (See page 2-2)

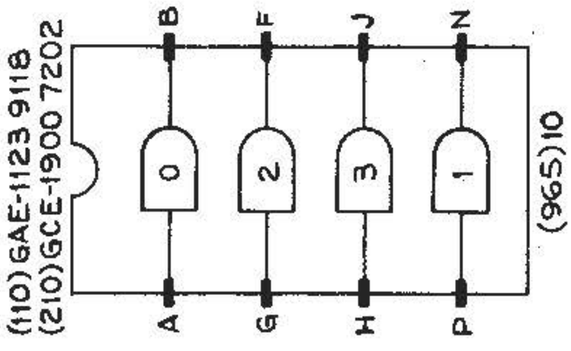
CTμL INTEGRATED CIRCUITS

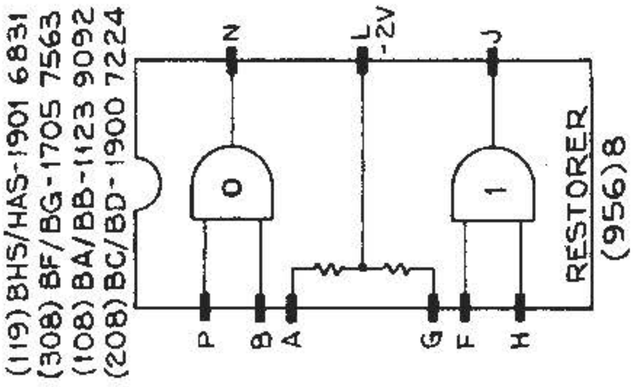
CTμL Voltage Levels:

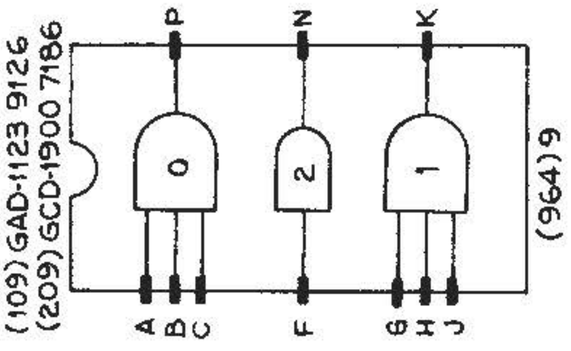


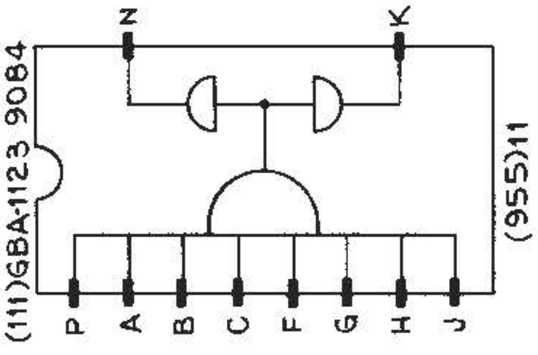
NOTE: Voltage levels at 27°C with maximum dc load.

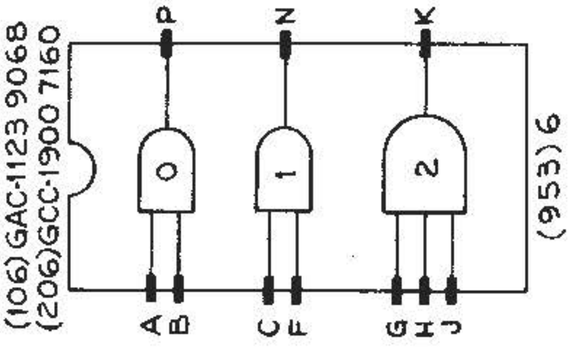
B 6700 INTEGRATED CIRCUIT CHIPS

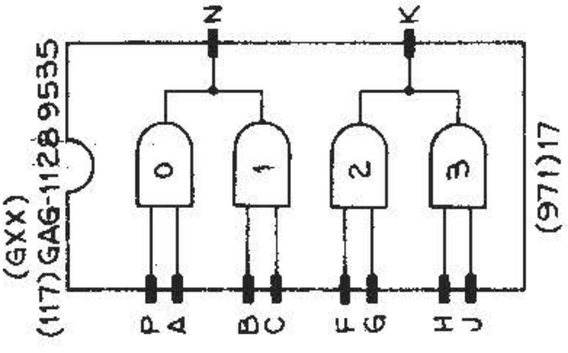


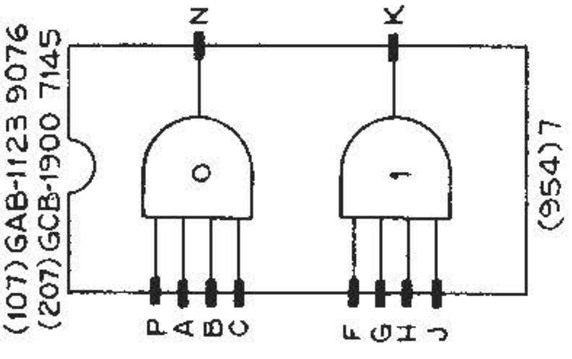


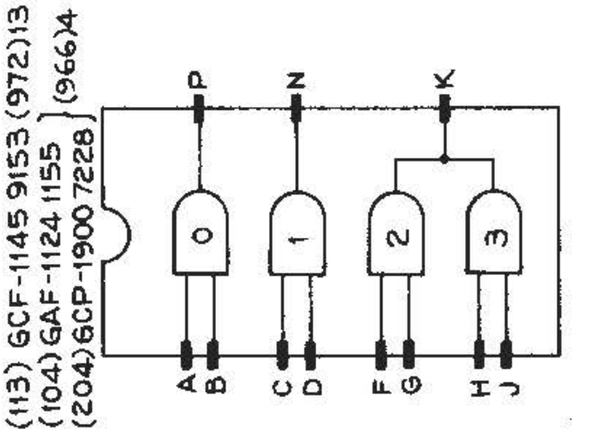






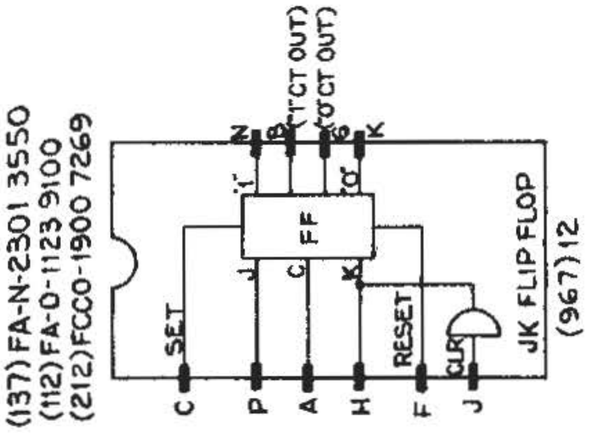


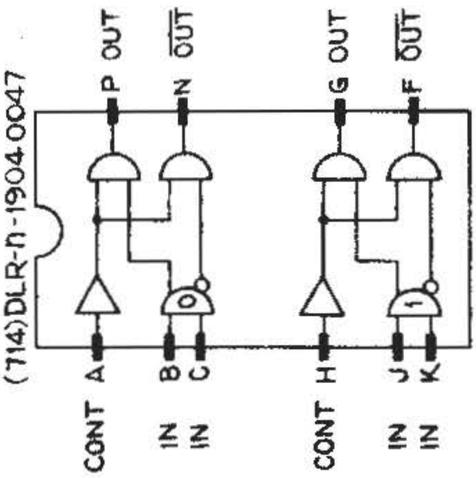


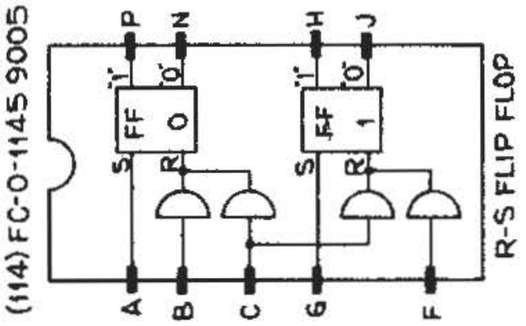


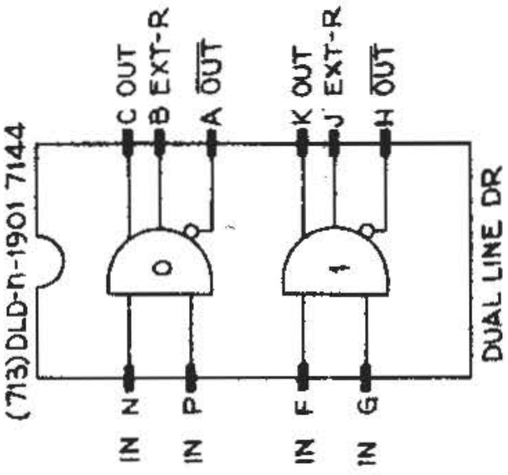
CTμL INTEGRATED CIRCUITS

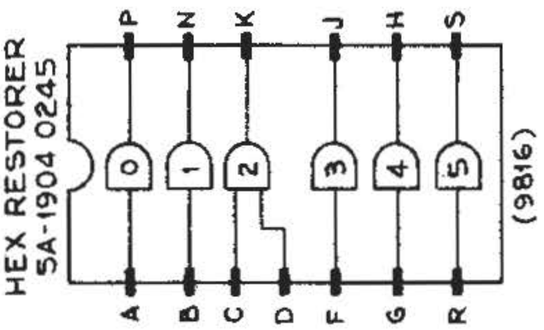
B 6700 Integrated Circuit Chips (Cont)

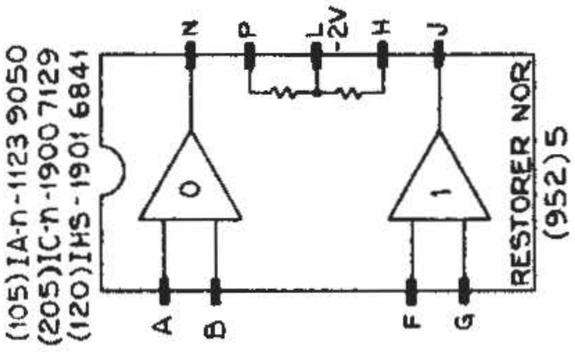


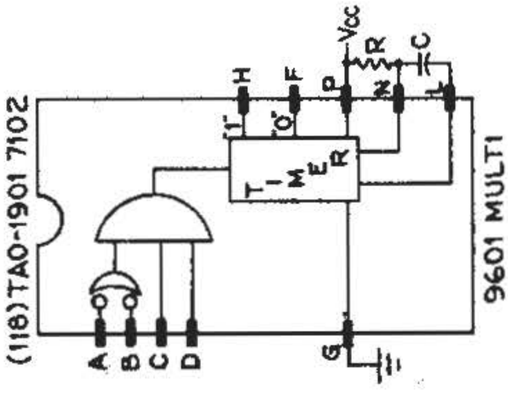




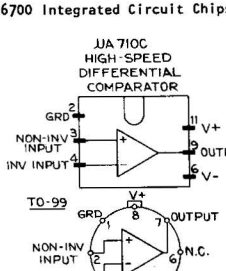


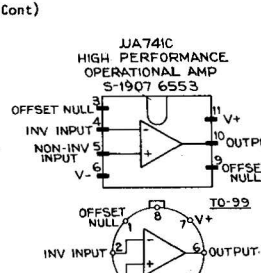


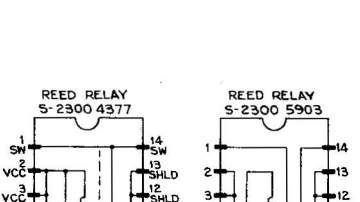




B 6700 Integrated Circuit Chips (Cont)







5. Q = TRUE OUTPUT; $\overline{Q}$ = FALSE OUTPUT; C = CLOCK.

4. ALL CHIPS SHOW TOP VIEW.

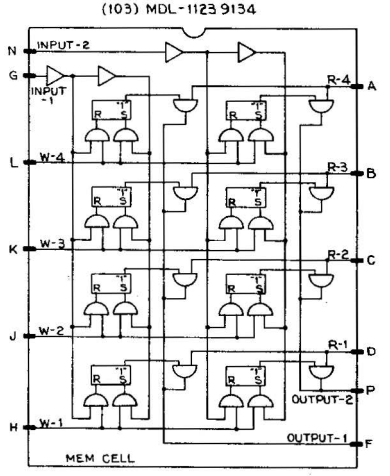
3. MARK (A) DENOTES FAIRCHILD INTEGRATED CKT.

2. GCF'S ARE NOT INTERCHANGEABLE WITH GAF'S OR GCP'S.

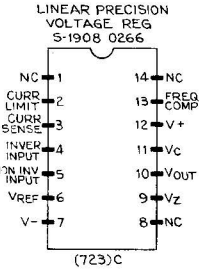
1. "C" TYPE CHIPS (200 SERIES MACH NO.S ARE REPLACEABLE WITH TYPE "A"S (100 SERIES MACH NO.S) BUT "A" TYPES ARE NOT REPLACEABLE WITH TYPE "C"S.

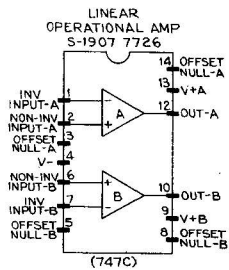
CTμL INTEGRATED CIRCUITS

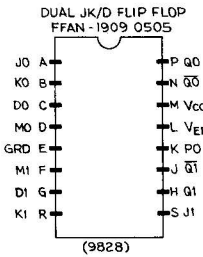
B 6700 Integrated Circuit Chips (Cont)



B 6700 Integrated Circuit Chips (Cont)







INDEX OF FINAL ACCEPTANCE PROGRAMS

MAIN FRAME AND PERIPHERALS

Name	Specification Number
Card Punch and Card Reader	T-1913-5433
Core Memory	T-1913-5110
Disk File	T-1913-5516
Display Units System	T-1913-4956
Line Printer (Buffered and Unbuffered)	T-1913-5359
Magnetic Tape System	T-1913-5276
Multiplexor	T-1913-5037
Processor All Orders	T-1913-5193

DATA COMMUNICATIONS SUBSYSTEM

Name	Specification Number
Auto Call Procedure Routine	A-1917-3434
Back-Back Test	A-1917-3491
Bootstrap Local Memory (LM)/ Main Memory (MM) Loader	A-1917-3616
Can Test	A-1917-3509
Card Local Memory (LM)/ Main Memory (MM) Loader	A-1917-3590
Cluster All Orders	A-1917-5672
Cluster Test Routine	A-1917-3483
Data Modify Utility Routine	A-1917-3657
DCP All Orders Test Routine	A-1917-3525
Executive Controller	A-1910-6046
I/O Control	A-1917-3608

DATA COMMUNICATIONS SUBSYSTEM (Cont)

Name	Specification Number
Line Cluster Status Routine	A-1917-3517
Local Memory Test Routine	A-1917-3533
Main Memory (MM)/Local Memory (LM) Loader	A-1917-3624
Memory Modify Routine	A-1917-3582
Pre-enable and Disable Utility	A-1917-3665
Pseudo Control	A-1917-3475
Tape-Disk Loader	A-1917-3640
TC500	A-1917-3442
"Teletype" Procedure Routine	A-1917-3426
"Touch-Tone" Voice Response	A-1917-3459