



ABSTRACT

It is a distinct advantage to have predictable real-time data against which to compare the behavior of new real-time programs. The following report describes a system used to check out programs written to decommutate and process telemetry (TM) data in real-time. Although the system is admittedly special purpose, it is believed that the principles used may have wide application in real-time simulation.

The system hardware consists of a general purpose digital computer with interrupt hardware, a TM data processor, and an inexpensive interrupt control and interface buffer. System software consists of a real-time TM data monitor and a conversational-mode interpreter used to construct data lists. In use, the telemetry data is generated by the computer and "transmitted" to the TM processor via the interface.

A SOFTWARE PACKAGE FOR REAL-TIME SIMULATION

Rationale & Problem Description

Real-time program check out can proceed only to a certain point before applying real-time constraints. Depending on a number of factors, this point can range from very early to very late in the debugging process. Usually the most critical portions of the program reside in that portion which requires a real-time situation similar to the real-time specifications of the program, to adequately verify that the program is working as intended. This is particularly true where the program is processing incoming data which is entering at a fixed but uncontrolled rate (i.e., uncontrolled by the program) and output equipment is slow compared to the input rate.

A particular example of such a program is one designed to decommutate and process telemetered data which is in the form of a serial bit stream. Such a program was written for the Apollo Lunar Surface Experiments Package (ALSEP). Specifically, the program (actually a group of programs) is required to acquire sync, decommutate, process, and output data to a very high speed line printer, or a digital-to-analog converter. In ALSEP, data is generated, converted to digital form and commutated before transmission. Some of this data is sub-commutated to as many as three lower levels. Implied in these statements is the presence of a sync-code for each commutation or "frame". Total synchronization was quite complex.

The programming assignment was eased somewhat by the use of a fairly advanced telemetry data processor. However, since it was a new



machine and the programming staff initially unfamiliar with its operation, adequate debugging aids became proportionally more important.

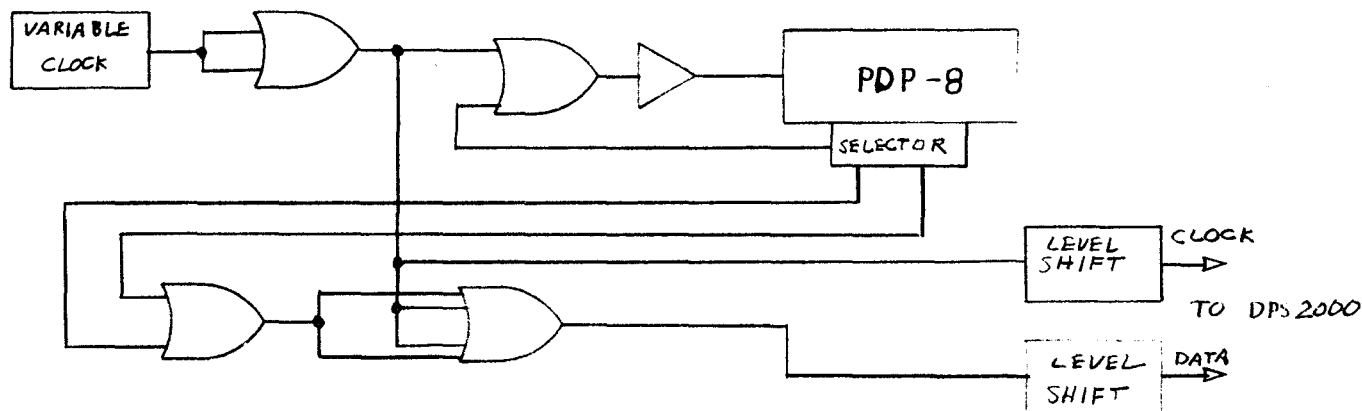
Due to the imposed bit rates, the required processing and the available hardware and software, a real-time operating system was written. This included a real-time input/output executive, subroutines for acquiring main frame sync, arithmetic processes, and printing. The total ALSEP telemetry format was partitioned into its experiments. A program was written for each experiment; that program was responsible for sync acquisition and processing peculiar to that experiment.

It became apparent fairly early in the project that some means to accurately simulate the ALSEP data format was needed. It was determined that these needs could best be fulfilled by using a small general purpose computer with interrupt hardware and a simple interrupt control/interface buffer. A software package was written for the simulating computer. It is this package that is the major topic of this report.

BASIC STRUCTURE

HARDWARE

1. Telemetry Data Processor: DPS 2000, manufactured by Bendix Electrodynamics Division. This is a special purpose digital computer with a section devoted entirely to telemetry sync acquisition and decommutation. The instruction set is divided about equally among TM and typical data processing commands. The particular version used has a 32 bit word, 4K magnetic core storage and a $2\mu s$ cycle time. It is capable of handling TM bit rates of up to 10^6 B. P. S. I/O equipment consists of a 15 cps typewriter, a 6000 LPM line printer and a 300 cps paper tape reader. There was also an eight channel D to A converter feeding a corresponding strip chart recorder.
2. Simulating Computer: PDP-8, manufactured by Digital Equipment Corporation. This is a 12 bit, 4K machine with a $1.5\mu s$ cycle time. The I/O configuration consisted of an ASR 33 teletype only.
3. Interrupt Control/Interface Buffer: The circuit is shown below.





SOFTWARE

1. Monitor: The purpose of this program is to thread through data lists and set up a waiting buffer from which an interrupt routine can draw data in real-time for output to the telemetry processor. The monitor is also capable of modifying the data stream in real-time via the teletype as the data is loaded into the waiting buffer. These modifications consist of (1) permanent changes to any part of the data, (2) temporary changes to any part of the data, (3) temporary changes to the main frame sync code, (4) addition of bits to the main frame for one MF time only, and (5) deletion of bits from the main frame for one MF time only. The monitor is capable of subframing to any level within core storage constraints. It is capable of handling fixed data, incrementing or decrementing count data, and random data. It can accommodate word lengths from 1 to 10 bits and has provision for three "special cases" (e.g., parity computation for a specific sub-set of words in the main frame list).

2. Interpreter: The purpose of this program is to construct the data lists used with the monitor. This is a conversational-mode program that interprets user responses to computer-generated requests for information and punches a binary paper tape which can be loaded into the PDP-8 along with the monitor. It sets up the data lists to be used by the monitor. It has provision for decimal or octal data input, and multiple addressing of a given subframe list. Certain shortcuts are available for inputting data. There is an error recovery facility available for the convenience of the user. Upon completing the input of data, the data lists are punched on paper tape in binary, with a checksum. Frame headers, counter addresses, and the data lists are printed.

SIMULATOR THEORY & IMPLEMENTATION

The algorithm used in the simulation treated the test data as if it were actually being commutated. The data is arranged in circular lists, each list having a pointer (or commutator arm to extend the analogy). Since ten bits was a convenient maximum simulator word size, two bits remain unused in the PDP-8 word. These two bits are used as flag bits to inform the monitor that the list position being presently examined is a switch to a lower level of commutation.

Since it was desired to have various types of data, as many as three special cases, and various word sizes from 1 to 10 bits, it was necessary to carry a descriptor word along with each data word. This is where maximum core economy had to be exercised. Without any effort to pack information efficiently, the total ALSEP simulator data would have occupied nearly all



the available core, leaving virtually no room for the monitor. Each descriptor word was therefore cut to 6 bits; two for special cases, two for data type, and two for word size. To accommodate a bit count from 1 to 10 the following approach was taken. The two word size bits are tested for zero. If they are not zero, 7 is added to give the size of the current word. If they are equal to zero, bits 2^9 , 2^8 , 2^7 of the data word are extracted and right justified to become the current word size. Data type flags are used to switch control to the appropriate routine. Count data in the data list consists of a low core base address where the pointer, IP & RP are stored. Random data is generated and the appropriate number of bits extracted as data.

Alternating waiting buffers are used to allow the output routine to draw its data from a completely filled buffer. These buffers each accommodate one main frame. Data for the next main frame is always being generated by the list processor. When the next frame is complete, control transfers to an idle loop. At the completion of the "transmission" of the current frame, the buffers are switched and the construction of the following frame is started.

SOFTWARE DETAIL

Monitor:

The monitor may be divided into three logically distinct sections:

(1) A list processing routine: The prime function is to thread through the data lists, constructing a main frame list in one of two alternating buffers for later output.

(2) An interrupt service routine (ISR) including output routine: The prime function of the ISR is to respond to an interrupt, and identify it as an interface, TTY input, or TTY output interrupt, and route program control to the appropriate routine. The output routine sequences through the appropriate buffer, "transmitting" one bit of data per interrupt.

(3) A real-time modifications routine controls the execution of modifications via TTY commands.

A commented assembly listing follows.

+1

0001	5402		04P 1 INTR	
0002	1621	INTR,	1621	/INTERRUPT SERVICE ROUTINE
		AR1=	10	/INDEX REGISTER #1
		AR2=	11	/INDEX REGISTER #2
		AR4=	13	/INDEX REGISTER #4
		AR5=	14	/INDEX REGISTER #5
		TBITS=	20	/TOTAL # OF BITS IN MF
		BASEAD=	21	/BASE ADDR OF MF LIST
				*22
0022	0000	CONT,	0	/NO. OF CONTROL CHARS
0023	0000	CHAR1,	0	/RTM
0024	0200	LTBL1,	200	/ADDR OF ONE OUTPUT TABLE
0025	0301	LTBL2,	301	/ADDR OF OTHER OUTPUT TABLE
0026	0000	AFIND,	0	/EVEN-ODD COUNTER FOR TABLE SELECTION
0027	0000	TABLE1,	0	/ONE OUTPUT TABLE
0030	0000	TABLE2,	0	/OTHER OUTPUT TABLE
0031	0000	BITPRD,	0	/BITS PER WORD FOR OUTPUT (PRESENTLY)
0032	0000	NBITS,	0	/MINUS TBITS
0033	0000	NOBITS,	0	/NO. OF BITS IN GENERATED WORD
0034	0000	DATA1,	0	/DATA WORD (LIST B)
0035	0000	COUNT,	0	/TEMP LOCATION
0036	0000	BITS,	0	/TEMP LOC'N
0037	0000	DATA2,	0	/DISCRIPTOR WORD
0040	1232	SPCASE,	SCASE	
0041	1231	ARADNO,	RADNO	
0042	0600	LOUTPT,	OUTPUT	
0043	0421	LSTRT,	START	
0044	1234	LEND,	SEND	
0045	0000	PREVCT,	0	
0046	0000	SAVEAD,	0	
0047	0000	SAVEPL,	0	
0050	0000	SAVEPA,	0	
0051	0000	PREVW,	0	
0052	0000	OK,	0	
0053	0000	BARNO,	0	
0054	0000	NSENO,	0	
0055	1000	LPICKU,	PICKUP	
0056	0000	ADATA1,	0	/ADDR OF DATA WORD
0057	0000	LIMIT2,	0	/INIT POINT FOR COUNTERS
0060	0000	RNDATA,	0	/RNG DATA RETURN
0061	0000	SAVE01,	0	/DESCRIPTOR 6 BITS, SHFTD LEFT 3
0062	0000	COUNT1,	0	/TEMP LOC'N
0063	0000	LIMIT1,	0	/RESET POINT FOR COUNTER
0064	1710	LIDLE,	IDLE	
0065	1200	LFIXIT,	FIXIT	
0066	0000	PRVCT,	0	/RT 10 BITS OF DATA COUNT POINTER
0067	0000	SBIT,	0	/SEVEN OR LESS BIT FLAG
0070	2003	LRST1,	RESET1	
0071	2020	LRST2,	RESET2	
0072	2012	LRST5,	RESET5	
0073	2000	LRST3,	RESET3	
0074	0000	NOPTR,	0	
0075	0000	TEMP1,	0	
0076	0000	TEMP,	0	
0077	0000	CNT,	0	
0100	0000	ADDR,	0	
0101	0000	ADDR1,	0	

01

05379

0102	0000	TEMP3,	0	
0103	7771	FIRST,	7771	/300-LAST
0104	0307	LAST,	307	
0105	7524	MCOMA,	7524	
0106	7563	ACR,	7563	
0107	0000	TEMPST,	0	
0110	0067	LSBIT,	SBIT	
0111	0057	LLMT2,	LLMT2	
0112	0066	LPRVCT,	PRVCT	
0113	0471	LINK01,	LINK01	
0114	0000	SAVEA,	0	
0115	0000	SAVEL,	0	
0116	0000	CHAR,	0	
0117	2131	LERROR,	ERROR	
0120	1443	ADTBL,	TBLU-301	
0121	2154	LR5BRK,	RSBRK	
0122	2146	LR5NSE,	RSNSE	
0123	0000	CHANGE,	0	
0124	0277	QUEST,	277	
0125	0000	ANSW,	0	
0126	0000	SAV01,	0	
0127	0000	FLAG,	0	
0130	2040	LNFLAG,	NFLAG	
0131	0144	LBFLAG,	BFLAG	
0132	2023	LNOW,	NOW	
0133	1313	LCONVT,	CONVT	
0134	1273	LRANGE,	RANGE	
0135	1057	LCHWRD,	CHWRD	
0136	0000	ACK,	0	
0137	0000	CHCONT,	0	
0140	7240		CLA CMA	
0141	1022		TAD CONT	
0142	3022		DCA CONT	/DECREASE NO. OF CONTROL CHAR BY 1
0143	5537		JMP I CHCONT	
0144	0000	BFLAG,	0	
0145	0000	AKEEP2,	0	

*402

0402	6032		KCC	
0403	6042		TCF	
0404	3052		DCA OK	
0405	3022		DCA CONT	/NO. OF CONTROLS
0406	3026		DCA AFIND	/COUNTER FOR ADDRESS IN BIT OUTPUT
0407	3530		DCA I LNFLAG	/NOISE FLAG
0410	3531		DCA I LBFLAG	/BARKER FLAG
0411	1064		TAD LIDLE	/IDLE LOOP, CHANGE FOR RTM
0412	3046		DCA SAVEAD	
0413	1177		TAD [1600]	
0414	3011		DCA XK2	
0415	1177		TAD [1600]	
0416	3010		DCA XR1	
0417	4465		JMS I LFIXIT	/RE-INIT ROUTINE
0420	6001		ION	
0421	7200	START,	CLA	/BEGIN MAIN LOOP
0422	1021		TAD BASEAD	/BASE ADDR OF MF
0423	4455		JMS I LPICKU	/RTN WITH ADD OF DATA WORD IN AC
0424	0000	ARG1,	0	/6 BITS OF DESCRIPTOR
0425	3056		DCA ADATA1	/ADDRESS OF FIRST DATA WORD
0426	1456		TAD I ADATA1	/FIRST DATA WORD
0427	7104		CLL RAL	
0430	7500		SMA	
0431	7430		SZL	
04				

05380

0438	5355	JMP LOWER /LOWER FRAME
0439	7313	RAR
0439	3034	DCA DATA1 /FIRST DATA WORD
0439	1224	TAD ARG1
0439	3037	DCA DATA2 /DESCRIPTOR
0437	1037	TAD DATA2
0440	7004	RAL /CHECK FOR SPECIAL CASES
0441	7420	SNL
0442	7513	SPA
0443	5140	JMP 1 SPCASE /SPECIAL CASE, DUMMY PRESENT
0444	3037	DCA DATA2 /SECOND DATA WORD SHIFTED LEFT ONE - 58
0445	1037	TAD DATA2
0446	7006	RTL /CHECK DATA TYPE
0447	7430	SZL
0450	5276	JMP IODCNT /INCR. OR DEC. COUNT
0451	7500	SMA
0452	5272	JMP LINK01+1 /FIXED PATTERN
0453	4441	JMS I ARADNO /CALL RANDOM NUMBER GENERATOR
		/ * * * * * ARADNO IS A DUMMY ADDRESS ON PAGE ZERO
		/ **** CHECK ARGUMENTS
		/ **** CHECK ARGUMENTS
0454	3060	DCA RNDATA /DATA FROM RANDOM NUMBER GENERATOR
0455	1037	TAD DATA2 /SECOND DATA WORD
0456	7006	RTL /CHECK NO.
0457	7006	RTL
0460	7530	SPA SZL /SKIP IF AC GE 0 AND LINK = 0
0461	5274	JMP GT7BIT /MORE THAN 7 BITS OF DATA
0462	7200	CLA /LESS THAN 7 BITS OF DATA
0463	1060	TAD RNDATA /RANDOM DATA
0464	0176	AND [177] /MASK OFF LAST 7 BITS
0465	3060	DCA RNDATA
0466	1034	TAD DATA1 /PICK UP NO. OF BITS
0467	0177	AND [1600] /IN BITS 2-4 OF DATA WORD
0470	1060	TAD RNDATA
0471	3034	LINK01, DCA DATA1
0472	4442	JMS I LOUPT /JUMP TO OUTPUT SUBROUTINE
0473	5221	JMP START
0474	7200	GT7BIT, CLA
0475	5270	JMP LINK01-1
0476	3061	IODCNT, DCA SAVE01 /INC. OR DEC. FLAG, DATA 2(DESCRIPTOR)SH
0477	1037	TAD DATA2 /LEFT SHFTD ONE
0500	7006	RTL
0501	7006	RTL
0502	7530	SPA SZL /TEST WORD SIZE
0503	7120	STL /SET LINK IF LINK OR SIGN=1
0504	7200	CLA
0505	1434	TAD I DATA1 /COUNTER POINTER
0506	0175	AND [1777] /STRIP NZIP FLAG
0507	3066	DCA PRVCT /COUNT DATA
0510	1066	TAD PRVCT
0511	7420	SNL
0512	0176	AND [177] /YES, STRIP BIT COUNT
0513	3035	DCA COUNT
0514	7004	RAL
0515	3067	DCA SBIT /SAVE FLAG
0516	1434	TAD I DATA1 /IF NEG. RP LOCATION IS ADDR
0517	7500	SMA
0520	5337	JMP CHECK
0521	7200	CLA
05		

05381

0522	1034		TAD DATA1
0523	3012		DCA AR3 /ADDRESS OF POINTER
0524	1412		TAD I AR3 /RESET POINT
0525	3063		DCA LIMIT1
0526	1412		TAD I AR3 /INIT. POINT
0527	3057		DCA LIMIT2
0530	1035		TAD COUNT /VALUE OF POINTER
0531	7041		CIA
0532	1063		TAD LIMIT1 /RESET POINT-POINTER
0533	7450		SNA
0534	5470		JMP I LRST1 /SET POINTER = INIT. POINT
0535	7510		SPA
0536	5471		JMP I LRST2 /DEC. COUNTER
0537	7200	CHECK,	CLA /INCR COUNTER 0/10 AND 2/END
0540	1034		TAD DATA1 /LIST 3
0541	3012		DCA AR3
0542	3057		DCA LIMIT2
0543	1412		TAD I AR3 /RESET POINT
0544	7041		CIA
0545	1035		TAD COUNT /POINTER-RESET POINT
0546	7440		SZA /SKIP IF READY FOR RE-CYCLE
0547	5473		JMP I LRST3 /INCREMENT COUNTER
0550	1067		TAD SBIT
0551	7450		SNA
0552	5470		JMP I LRST1
0553	7200		CLA
0554	5472		JMP I LRST5
0555	7510	LOWER,	RSA
0556	5223		JMP START+2
			= 600
0600	0000	OUTPUT,	0
0601	7200		CLA
0602	1037		TAD DATA2 /SECOND DATA WORD, 6 BITS SHFTD LFT 1
0603	7006		RTL /RIGHT JUSTIFY COUNT
0604	7006		RTL
0605	7006		RTL /SHIFT SIZE INTO LINK AND AC
0606	0174		AND L3J /MASK OFF WORD LENGTH
0607	7450		SNA
0610	5214		JMP LTEST /LESS THAN 8 BITS
0611	1173		TAD L7J
0612	3035		DCA COUNT /BIT COUNT
0613	5222		JMP COMMON+1
0614	1034	LTEST,	TAD DATA1 /FIRST DATA WORD
0615	7006		RTL
0616	7006		RTL
0617	7006		RTL /LEFT ADJUST DATA IN DATA WORD
0620	0173		AND L7J /MASK OFF BIT COUNT
0621	3035	COMMON,	DCA COUNT /BIT COUNT
0622	7200		CLA
0623	1035		TAD COUNT
0624	1172		TAD I-12J /DEC 10
0625	7450		SNA
0626	5235		JMP COMM1 /IF 10 BITS
0627	3747		DCA I LCNT /ODD-EVEN STORE
0630	1034		TAD DATA1
0631	7104		CLL RAL
0632	2747		ISZ I LCNT
0633	5231		JMP *-2
06			

05382

0634	3034		DCA	DATA1
0635	1034	COMN1,	TAD	DATA1
0636	7006		RTL	
0637	3034		DCA	DATA1 /LEFT JUSTIFIED COUNT
0640	1035		TAD	COUNT /NO. OF BITS
0641	7041		CIA	
0642	3747		DCA	I LCNT
0643	7120		STL	/SET UP MASK
0644	7010		RAR	
0645	2747		ISZ	I LCNT /NEEDED DATA
0646	5243		JMP	--3
0647	0034		AND	DATA1 /WITH LEFT JUST. COUNT
0650	3034		DCA	DATA1 /MASKED DATA
0651	1045		TAD	PREVCT /NO. OF BITS IN OUTPUT WORD
0652	7450		SNA	
0653	5345		JMP	SETIN
0654	7041		CIA	
0655	3747		DCA	I LCNT
0656	1034		TAD	DATA1
0657	7110		CLL	RAR /SHIFT DATA WORD 1
0660	2747		ISZ	I LCNT /INTO PROPER POSITION
0661	5257		JMP	--2
0662	0171		AND	[7774] /MASK OFF BITS 0-9; ONLY 10 BITS IN WORD
0663	1427		TAD	I TABLE1
0664	3427	SETIN1,	DCA	I TABLE1 /CRNT ADDR IN PRES TBL
0665	1172		TAD	[-12]
0666	1045		TAD	PREVCT /NO. OF BITS IN OUTPUT WORD - 10
0667	3747		DCA	I LCNT /TEMPORARY STORAGE
0670	1045		TAD	PREVCT
0671	1035		TAD	COUNT /NO. OF BITS ENTERING PRESENT CURRNT 0
0672	3045		DCA	PREVCT /UPDATE NO. OF BITS IN OUTPUT WORD
0673	1045		TAD	PREVCT
0674	1172		TAD	[-12]
0675	7510		SPA	
0676	5312		JMP	OUT+2 /TOTAL NO. OF BITS .LT. 10
0677	3045		DCA	PREVCT /RESET TOTAL NO. OF BITS
0700	1045		TAD	PREVCT
0701	7450		SNA	
0702	5310		JMP	OUT /TOTAL # OF BITS = 10
0703	7200		CLA	
0704	1034		TAD	DATA1
0705	7104		CLL	RAL /SHIFT REMAINING BITS
0706	2747		ISZ	I LCNT /FOR NEXT OUTPUT WORD
0707	5305		JMP	--2
0710	2027	OUT,	ISZ	TABLE1 /UPDATE ADDRESS OF OUTPUT WORD
0711	3427		DCA	I TABLE1
0712	7200		CLA	/MUST BE IN FOR PREVIOUS JUMP *****
0713	1036		TAD	BITS /CHECK IF THROUGH GENRTG THIS MF
0714	1035		TAD	COUNT
0715	3036		DCA	BITS /UPDATE NO. OF BITS PROCESSED
0716	1036		TAD	BITS
0717	7041		CIA	
0720	1020		TAD	TBITS
0721	7540		SMA	SZA
0722	5443		JMP	I LSTRT /CONTINUE LOOP
0723	7200	IDLE1,	CLA	
0724	1530		TAD	I LNFLAG /NOISE FLAG & TEST CNTR
0725	7450		SNA	
0726	5332		JMP	CHBRK /CHECK BARKER CODE
0727	2054		ISZ	NSEN0
0730				

05383

0730	5332		JMP CHBRK	
0731	4522		JMS I LRSNSE	
0732	1531	CHBRK,	TAD I LBFLAG	/CHECK BARKER CODE FLAG
0733	7450		SNA	
0734	5340		JMP IDLE2	
0735	2053		ISZ BARNO	/NO. OF TIMES THRU LOOP
0736	5340		JMP IDLE2	
0737	4521		JMS I LRSBRK	/RESTORE B FLAG, ETC
0740	7200	IDLE2,	CLA	
0741	1047		TAD SAVEPL	
0742	7110		CLL RAR	
0743	1050		TAD SAVEPA	
0744	5446		JMP I SAVEAD	/RTM (IDLE)
0745	1034	SETIN,	TAD DATA1	
0746	5264		JMP SETIN1	/STORE IN GEN. TABLE
0747	0077	LCNT,	CNT	
				*1000
			/ADDRESS OF WHOLE WORD- $2(N+1)+BA+P(N)$	
			/ADDRESS OF HALF WORD- $2(N+1)+BA+RP(N)+1+P(N)/2$	
			/WHERE $N=0,1,2,3,4$ -BA=ADDRESS OF $P(N)$	
1000	0000	PICKUP,	0	/OBTAINS DATA WD ADDR AND (DESCRIPTOR WD)
1001	3100		DCA ADDR	/ADDRESS OF $P(N)$
1002	7201		CLA IAC	
1003	1100		TAD ADDR	
1004	3101		DCA ADDR1	/ADDR OF $RP(N)$
1005	1501		TAD I ADDR1	
1006	7106		CLL RTL	
1007	7004		RAL	
1010	7001		IAC	
1011	7104		CLL RAL	
1012	0170		AND [17]	/2(N+1)
1013	1100		TAD ADDR	
1014	3074		DCA NOPTR	/2(N+1)+BA
1015	1501		TAD I ADDR1	
1016	0175		AND [1777]	
1017	3076		DCA TEMP	/RP(N)
1020	1500		TAD I ADDR	/P(N)
1021	7110		CLL RAR	
1022	7001		IAC	
1023	1076		TAD TEMP	/1+RP(N)+P(N)/2
1024	1074		TAD NOPTR	/2(N+1)+BA+1+RP(N)+P(N)/2
1025	3075		DCA TEMP1	/ADDRESS OF HALF WORD
1026	1475		TAD I TEMP1	
1027	7420		SNL	
1030	5234		JMP .+4	
1031	7006		RTL	/ROTATE RIGHT SIX BITS
1032	7006		RTL	
1033	7006		RTL	
1034	0377		AND (7700)	/HALF WORD RIGHT JUSTIFIED
1035	3600		DCA I PICKUP	/ARGU. ADDRESS
1036	2200		ISZ PICKUP	
1037	1074		TAD NOPTR	/2(N+1)+BA
1040	1500		TAD I ADDR	/2(N+1)+BA+P(N)
1041	3074		DCA NOPTR	/ADDRESS IN LIST
1042	1500		TAD I ADDR	
1043	7041		CIA	
1044	1076		TAD TEMP	/RP(N)-P(N)
1045	7450		SNA	
1046	5251		JMP RTN	/IF AC=0, P(N)=0, IF AC NOT 0, P(N)=P(N)+1
104				

05384

1047	7201		CLA IAC	
1050	1500		TAD I ADDR	
1051	3500	RTN,	DCA I ADDR	
1052	1074		TAD NOPTR	
1053	5600		JMP I PICKUP	
1054	4532	WRDCH,	JMS I LNOW	
1055	4257		JMS CHWRD	
1056	5464		JMP I LIDLE	
1057	0000	CHWRD,	0	
1060	4533		JMS I LCONVT	/CONVERT BCD TO OCTAL
1061	3123		DCA CHANGE	
1062	1523		TAD I CHANGE	
1063	3051		DCA PREVW	
1064	4137		JMS CHCONT	
1065	1022		TAD CONT	
1066	7450		SNA	
1067	5265		JMP --2	/WAIT FOR CONTROL CHAR
1070	4533		JMS I LCONVT	
1071	3523		DCA I CHANGE	/INSERT CORE CHANGE
1072	4137	REDO,	JMS CHCONT	
1073	1136		TAD AOK	
1074	7440		SZA	
1075	5657		JMP I CHWRD	
1076	1022		TAD CONT	
1077	7450		SNA	
1100	5276		JMP --2	/WAIT FOR CONTROL CHAR
1101	1011		TAD XR2	
1102	3126		DCA SAV01	
1103	1526		TAD I SAV01	/LOAD NEXT CHAR
1104	4534		JMS I LRANGE	
1105	7440		SZA	
1106	5657		JMP I CHWRD	
1107	2123		ISZ CHANGE	/CHANGE LOCATION
1110	4533		JMS I LCONVT	
			DCA I CHANGE	/CHANGE CO-E LOCAT+0N2
1111	3523		JMP REDO	
1177	7700			*1200
1200	0000	FIXIT,	0	
1201	2026		ISZ AFIND	
1202	1026		TAD AFIND	
1203	7010		RAR	
1204	7200		CLA	
1205	7430		SZL	
1206	5213		JMP SETBUF	
1207	1024		TAD LTBL1	/ADDRESS OF TABLE1
1210	3027		DCA TABLE1	/TABLE FOR GENERATING BITS
1211	1025		TAD LTBL2	/ADDRESS OF TABLE 2
1212	5216		JMP COM1	
1213	1025	SETBUF,	TAD LTBL2	/ADDRESS OF TABLE2
1214	3027		DCA TABLE1	/TABLE FOR GENERATING BITS
1215	1024		TAD LTBL1	/ADDRESS OF TABLE1
1216	3030	COM1,	DCA TABLE2	/TABLE FOR OUTPUTTING BITS
1217	1020		TAD TBITS	/TOTAL NUMBER OF BITS
1220	7041		CIA	
1221	3032		DCA NBITS	/RESET NUMBER OF BITS TO BE PROCESSED
1222	3036		DCA BITS	
1223	1172		TAD [-12]	
1224	3031		DCA BTPWRD	/RESET BITS PER WORD COUNTER
1225	3045		DCA PREVCT	
1226	1172		TAD [-12]	

05385

```

1227 3033      DCA NBITS
1230 5600      JMP I FIXIT
                      /RETURN FROM SEND
                      /AC=0 NORMAL RETURN
                      /AC=-1 OUTPUT OF TABLE COMPLETED
                      /      CHANGE TABLES

1231 5231  RADNO,  JMP .
1232 5232  SCASE,  JMP .
1233 5233      JMP .
1234 0000  SEND,   0
1235 7200      CLA
1236 1430      TAD I TABLE2  /DATA WORD READY FOR OUTPUT
1237 7004      RAL
1240 3430      DCA I TABLE2
1241 1377      TAD (6471)  /GENERATE BIT OUTPUT INSTRUCTION
1242 7420      SNL  /OP CODE 6471=SEND 1611
1243 7001      IAC  /OP CODE 6472=SEND 0 BIT
1244 3245      DCA .+1
1245 0000      0  /GENERATED INSTRUCTION
1246 2031      ISZ BTPWRD  /BITS PER WORD FLAG
1247 5266      JMP INCTBT
1250 3430      DCA I TABLE2  /UPDATE DATA COUNTER-ALL BITS IN
1251 2030      ISZ TABLE2  /PRESENT DATA WORD PROCESSED
1252 1172      TAD I-121  /RE-INIT BITS PER WORD COUNTER
1253 3031      DCA BTPWRD
1254 1032      TAD NBITS  /CA NEG #) BITS OUTPUTTED IN THIS FR
1255 1127      TAD FLAG  /CHANGE LENGTH OF MF
1256 7500      SMA
1257 5265      JMP NOTNOW  /DO NOT ADD OR DELETE @ END OF MF
1260 3032      DCA NBITS  /SET BIT COUNT
1261 1127      TAD FLAG
1262 1031      TAD BTPWRD
1263 3031      DCA BTPWRD  /ADD OR DELETE BITS IN NEXT WORD
1264 3127      DCA FLAG  /CLEAR ADD-DEL QTY
1265 7200  NOTNOW,  CLA
1266 2032  INCTBT,  ISZ NBITS
1267 5634      JMP I SEND  /NORMAL EXIT-MORE BITS IN TABLE
1270 4200      JMS FIXIT
1271 7201      CLA IAC
1272 5634      JMP I SEND
1273 0000  RANGE,  0  /CHAR IN AC DIGIT
1274 3126      DCA SAV01  /CHAR MUST BE BETWEEN 260 AND 267
1275 1126      TAD SAV01
1276 1376      TAD (-270)
1277 7510      SPA
1300 5304      JMP .+4
1301 7200  OUTOL,  CLA
1302 1126      TAD SAV01
1303 5673      JMP I RANGE
1304 7200      CLA
1305 1126      TAD SAV01
1306 1375      TAD (-260)
1307 7510      SPA
1310 5301      JMP OUTOL
1311 7200      CLA
1312 5673      JMP I RANGE
1313 0000  CONV,   0  /CONVERT BCD TO OCTAL
1314 7200      CLA
1315 3125      DCA ANSW  /ANSWER
1316 1411  LOOP1,  TAD I XR2

```

1317	3126		DCA SAV01
1320	1011		TAD XR2
1321	0374		AND (1717)
1322	3011		DCA XR2
1323	1126		TAD SAV01
1324	4273		JMS RANGE
1325	7450		SNA
1326	5342		JMP INTOL
1327	1373		TAD (-212)
1330	7450		SNA
1331	5352		JMP BYCR
1332	7200		CLA
1333	1126		TAD SAV01
1334	1372		TAD (-254)
1335	7440		SZA
1336	5517		JMP I LERROR
1337	7200	CMPT,	CLA
1340	1125		TAD ANSW
1341	5713		JMP I CONVT
1342	1125	INTOL,	TAD ANSW
1343	7104		CLL RAL
1344	7104		CLL RAL
1345	7104		CLL RAL
1346	1126		TAD SAV01
1347	1375		TAD (-260)
1350	3125		DCA ANSW
1351	5316		JMP LOOP1
1352	7201	BYCR,	CLA IAC
1353	1011		TAD XR2
1354	0374		AND (1717)
1355	3011		DCA XR2
1356	5337		JMP CMPT
1372	7524	*1621	
1373	7566		
1374	1717		
1375	7520		
1376	7510		
1377	6471		
1621	3114		DCA SAVEA /SAVE AC
1622	7004		RAL
1623	3115		DCA SAVEL /SAVE LINK
1624	6041		TSF /SKIP ON TTY OUTPUT INTERRUPT
1625	5227		JMP .+2
1626	5304		JMP CLEAR /CLEAR TTO FLAG
1627	6031		KSF
1630	5270		JMP BITOUT /CLOCK INTERRUPT
1631	6036		KRB /TTY INPUT INTERRUPT
1632	3116		DCA CHAR /BRING IN CHAR FROM TTY
1633	1052		TAD OK
1634	7440		SZA
1635	5253		JMP RSTOR
1636	7201		CLA IAC
1637	3052		DCA OK
1640	1116		TAD CHAR
1641	6046		TLS
1642	1105		TAD MCOMA
1643	7440		SZA /COMPARE WITH COMMA
1644	5261		JMP CHCR
1645	2022	COMA,	ISZ CONT /NO. OF CONTROL CHAR

05387

1646	1116		TAD	CHAR
1647	3410		DCA	I XR1 /INSERT CHAR IN TABLE
1650	1010		TAD	XR1 /TABLE IS CIRCULAR
1651	0377		AND	(1717) /XR1 MUST BE 1600
1652	3010		DCA	XR1 /THRU 1617
1653	7203	RSTOR,	CLA	
1654	1115		TAD	SAVEI /RESTORE AC AND LINK
1655	7110		CLL	RAR
1656	1114		TAD	SAVEA
1657	6001		ION	
1660	5400		JMP	I 0 /INTERRUPT COMPLETED, RETURN
1661	7200	CHCR,	CLA	
1662	1116		TAD	CHAR
1663	1106		TAD	MCR /CHECK FOR CARRIAGE RETURN
1664	7450		SNA	
1665	5245		JMP	COMA /CONTROL CHAR
1666	7200		CLA	
1667	5246		JMP	COMA+1 /NON-CONTROL CHAR
1670	4444	BITOUT,	JMS	I LSEND /OUTPUT BIT TO DPS-2000
1671	7450		SNA	/AC=0 NORMAL EXIT
1672	5253		JMP	RSTOR /AC=1 OUTPUT TABLE CHANGED
1673	7203		CLA	
1674	1000		TAD	0
1675	3046		DCA	SAVEAD /SAVE ADDRESS OF INTERRUPT
1676	1115		TAD	SAVEI
1677	3047		DCA	SAVEPL
1700	1114		TAD	SAVEA
1701	3050		DCA	SAVEPA
1702	6001		ION	
1703	5443		JMP	I LSTRT
1704	6042	CLEAR,	TCF	
1705	7203		CLA	
1706	3052		DCA	OK
1707	5253		JMP	RSTOR
1710	4532	IDLE,	JMS	I LNOW
1711	7200	PHASE1,	CLA	
1712	1411		TAD	I XR2
1713	3023		DCA	CHAR1 /NEXT CHAR TO PROCESS
1714	7201		CLA	IAC
1715	1011		TAD	XR2
1716	0377		AND	(1717)
1717	3011		DCA	XR2 /SET XR2=XR2+1
1720	1411		TAD	I XR2
1721	1106		TAD	MCR /CHECK FOR CARRIAGE RETURN
1722	7440		SZA	
1723	5517		JMP	I LERROR /NO CARRIAGE RETURN
1724	4137		JMS	CHCONT
1725	1023	GO1,	TAD	CHAR1
1726	7041		CIA	
1727	1104		TAD	LAST /HIGHEST ASCII VALUE FOR CODE
1730	7510		SPA	
1731	5517		JMP	I LERROR
1732	1103		TAD	FIRST
1733	7500		SMA	
1734	5517		JMP	I LERROR
1735	7200		CLA	
1736	1023		TAD	CHAR1 /COMPUTE ADDRESS OF FUNCTION
1737	1120		TAD	ADTBL
1740	3102		DCA	TEMP3
1741	1502		TAD	I TEMP3
1742	3102		DCA	TEMP3
1743	5502		JMP	I TEMP3

05388

1744	1054	TBLC,	WRDCH	/"A"	
1745	1753		CV	/"B"	
1746	2031		NOISE	/"C"	
1747	1754		PTAPE	/"D"	
1750	2042		BARKER	/"E"	
1751	1755		ADDBTS	/"F"	
1752	1762		DELETE	/"G"	
1753	5353	CV,	JMP	.	
1754	5354	PTAPE,	JMP	.	
1755	4532	ADDBTS,	JMS	I LNOW	
1756	4533		JMS	I LCONVT	
1757	7041		CIA		
1760	3127		DCA FLAG	/CONTAINS # OF BTS TO ADD OR DELETE	
1761	5464		JMP	I LIDLE	
1762	4532	DELETE,	JMS	I LNOW	
1763	4533		JMS	I LCONVT	
1764	5360		JMP	.-4	
1777	1717				* 2000
2000	2434	RESET3,	ISZ	I DATA1	/INC. COUNTER-UPDATE IT
2001	7200		CLA		
2002	5214		JMP	RESET6	
2003	1510	RESET1,	TAD	I LSBIT	/SET POINTER = INIT. POINTER
2004	7010		RAR		
2005	7200		CLA		
2006	7430		SZL		
2007	5216		JMP	RESET4	
2010	1434		TAD	I DATA1	
2011	0377		AND	(5600)	
2012	1511	RESET5,	TAD	I LLMT2	
2013	3434	RESET7,	DCA	I DATA1	
2014	1512	RESET6,	TAD	I LPRVCT	/POINTER 10 BITS
2015	5513		JMP	I LLNK01	
2016	1376	RESET4,	TAD	(4000)	
2017	5212		JMP	RESET5	
2020	7240	RESET2,	CLA	CMA	/DEC. COUNT
2021	1434		TAD	I DATA1	
2022	5213		JMP	RESET7	
2023	0000	NOW,	0		
2024	7200		CLA		
2025	1022		TAD	CONT	
2026	7450		SNA		
2027	5225		JMP	.-2	
2030	5623		JMP	I NOW	
2031	4223	NOISE,	JMS	NOW	
2032	4533		JMS	I LCONVT	
2033	7041		CIA		
2034	3054		DCA	NSEN0	
2035	4256		JMS	SETCH	
2036	0000	KEEP1,	0		
2037	0000	AKEEP1,	0		
2040	0000	NFLAG,	0		
2041	5464		JMP	I LIDLE	
2042	4223	BARKER,	JMS	NOW	
2043	4533		JMS	I LCONVT	
2044	7041		CIA		
2045	3053		DCA	BARN0	
2046	7001		IAC		
2047	3144		DCA	BFLAG	
2050	7326		7326		
2051	1021		TAD	BASEAD	
2052	3145		DCA	AKEEP2	

05389

2053	7040		CMA	
2054	3545		DCA I AKEEP2	
2055	5464		JMP I LIDLE	
2056	0000	SETCH,	0	
2057	4137		JMS CHCONT	
2060	7201		CLA IAC	
2061	3136		DCA AOK	
2062	4223		JMS NOW	/CHECK FOR CONTROL CHAR IN BUFFER
2063	4535		JMS I LCHWRD	/CHANGE CORE WORD
2064	7200		CLA	
2065	3136		DCA AOK	
2066	1051		TAD PREVW	
2067	3656		DCA I SETCH	/SAVE OLD CORE WORD
2070	2256		ISZ SETCH	
2071	1123		TAD CHANGE	
2072	3656		DCA I SETCH	/ADDRESS OF CHANGED WORD
2073	2256		ISZ SETCH	
2074	7001		IAC	
2075	3656		DCA I SETCH	/SET FLAG
2076	2256		ISZ SETCH	
2077	5656		JMP I SETCH	
2100	0000	OUTCH,0		
2101	3107		DCA TEMPST	
2102	6001	HELP, ION		
2103	7000		NOP	
2104	6002		IOF	
2105	7200		CLA	
2106	1052		TAD OK	
2107	7440		SZA	
2110	5302		JMP HELP	
2111	7001		IAC	
2112	3052		DCA OK	
2113	1107		TAD TEMPST	
2114	6046		TLS	
2115	6001		ION	
2116	5700		JMP I OUTCH	
2117	0000	ENDL,	0	
2120	7200		CLA	
2121	1330		TAD CR	
2122	4300		JMS OUTCH	
2123	7200		CLA	
2124	1327		TAD LFEED	
2125	4300		JMS OUTCH	
2126	5717		JMP I ENDL	
2127	0212	LFEED,	212	
2130	0215	CR,	215	
2131	4317	ERROR,	JMS ENDL	/OUTPUT CARRIAGE RETURN AND LINE FEED
2132	7200		CLA	
2133	1124		TAD QUEST	/QUESTION MARK
2134	4300		JMS OUTCH	
2135	7200		CLA	
2136	4317		JMS ENDL	
2137	7200		CLA	
2140	3022		DCA CONT	/ZERO OUT NO. OF CONTROL CHAR
2141	1375	TAD	(1600)	
2142	3010		DCA XR1	
2143	1375	TAD	(1600)	
2144	3011		DCA XR2	
2145	5464		JMP I LIDLE	

05390

```

2146 0000 RSNSE,0
2147 7200 CLA
2150 1236 TAD KEEP1
2151 3637 DCA I AKEEP1/RESTORE VALUE
2152 3530 DCA I LNFLAG /SET NOISE FLAG = 0
2153 5746 JMP I RSNSE
2154 0000 RSBK,0
2155 1374 TAD (1611)
2156 3545 DCA I AKEEP2/FIRST TEN BITS OF BARKER CODE
2157 3144 DCA BFLAG /CLEAR BARKER CODE FLAG
2160 5754 JMP I RSBK
2174 1611 S
2175 1600
2176 4000
2177 5600
0170 0017
0171 7774
0172 7766
0173 0037
0174 0033
0175 1777
0176 0177
0177 1600

```

```

ADATA1 0056
ADDBTS 1755
ADDR 0100
ADDR1 0101
ADTBL 0120
AFIND 0026
AKEEP1 2037
AKEEP2 0145
ANS# 0125
AOK 0136
ARADNO 0041
ARG1 0424
BARKER 2042
BARNO 0053
BASEAD 0021
BFLAG 0144
BITOUT 1670
BITS 0036
BTPWMD 0031
BYCR 1352
CHANGE 0123
CHAR 0116
CHAR1 0023
CHBRK 0732
CHCONT 0137
CHCR 1661
CHECK 0537
CHWRD 1057
CLEAR 1704

```

05391

CMPT 1337
CNT 0077
COMA 1645
COMMON 0621
COMN1 0635
COM1 1216
CONT 0022
CONVT 1313
COUNT 0035
COUNT1 0062
CR 2130
CV 1753
DATA1 0034
DATA2 0037
DELETE 1762
ENDL 2117
ERROR 2131
FIRST 0103
FIXIT 1200
FLAG 0127
GO1 1725
GT7BIT 0474
HELP 2102
IDLE 1710
IDLE1 0723
IDLE2 0740
INCTBT 1266
INTOL 1342
INTR 0002
IODCNT 0476
KEEP1 2036
LAST 0104
LBFLAG 0131
LCHWRD 0135
LCNT 0747
LCONVT 0133
LERROR 0117
LFEEED 2127
LFIXIT 0065
LIDLE 0064
LIMIT1 0063
LIMIT2 0057
LINK01 0471
LLMT2 0111
LLNK01 0113
LNFLAG 0130
LNOW 0132
LOOP1 1316
LOUTPT 0042
LOWER 0555
LPICKU 0055
LPRVCT 0112
LRANGE 0134
LRSBRK 0121
LRSNSE 0122
LRST1 0070
LRST2 0071
LRST3 0073
LRST5 0072

05392

LSBIT 0110
LSEND 0044
LSTRT 0043
LTBL1 0024
LTBL2 0025
LTEGT 0614
MCOMA 0105
MCR 0106
NBITS 0032
NFLAG 2040
NOBITS 0033
NOISE 2031
NOPTR 0074
NOTNOW 1265
NOW 2023
NSEN0 0054
OK 0052
OUT 0710
OUTCH 2100
OUTOL 1301
OUTPUT 0600
PHASE1 1711
PICKUP 1000
PREVCT 0045
PREVW 0051
PRVCT 0066
PTAPE 1754
QUEST 0124
RADNO 1231
RANGE 1273
REDO 1072
RESET1 2003
RESET2 2020
RESET3 2000
RESET4 2016
RESET5 2012
RESET6 2014
RESET7 2013
RNDATA 0060
RSBRK 2154
RSNSE 2146
RSTOR 1653
RTN 1051
SAVEA 0114
SAVEAD 0046
SAVEL 0115
SAVEPA 0050
SAVEPL 0047
SAVE01 0061
SAV01 0126
SBIT 0067
SCASE 1232
SEND 1234
SETBUF 1213
SETCH 2056
SETIN 0745
SETIN1 0664
SPCASE 0040
START 0421
TABLE1 0027
TABLE2 0030

05393

TBITS	0020
TBLC	1744
TEMP	0076
TEMPST	0107
TEMP1	0075
TEMP3	0102
WRDCH	1054
XR1	0010
XR2	0011
XR3	0012
XR4	0013
XR5	0014

05394



4/20/67

Interpreter:

The interpreter may be divided into three logically distinct sections:

(1) A data list building section which operates in a conversational or question-and-answer mode. In this section data is inputted to the PDP-8 by the user via the teletype. When a data word is specified it is stored in appropriate locations. Frame list storage is automatically allocated. Control of the builder section is retained by this section until all of the data words have been accounted for. At that point control is passed to the output portion of the program which contains the remaining two sections.

(2) A punch section which punches a binary tape of the main frame description, counters and the data lists. A second block of data is punched which consists of some important registers used by the interpreter. These registers, when loaded over the interpreter, offer the user a convenient restart procedure.

(3) A dump section which lists the sizes and header locations of each frame list, a dump of the counters, and a dump of the data lists.

A commented assembly listing follows.

Prepared by:


C. Coleman

Approved by


F. G. Conklin
GSE Group Engineer

* 0

0000	7402		HLT
0001	1225	BASBAS,	MAT
0002	2547	MFBASE,	FIRST
0003	1000	RANDAD,	1000

* 5

0005	1135	ZLFLF,	LFLFLF	
0006	1377	CNTBAS,	1377	
0007	1116	ZDECPT,	DECPRT	
0010	1225	BASE,	MAT	
0011	1377	CNTADD,	1377	
0012	0000		0	/USED
0013	0000		0	/USED
0014	0000		0	/USED
0015	0000		0	/USED
0016	0000		0	/NOT USED
0017	0000		0	/USED BY MESSAGE
0020	0000	MONBIT,	0	/LOCATION IN MONITOR FOR # OF BITS
0021	2547	BASEAD,	FIRST	/LOCATION IN MONITOR FOR START OF MF
0022	1026	CONTRL,	TAD TABLE	
0023	0000	INPUT,	0	
0024	0000	NUMBER,	0	
0025	0020	COUNT,	0020	
0026	6340	TABLE,	-1440	/-800(10)
0027	7160		-0620	/-400
0030	7470		-0310	/-200
0031	7634		-0144	/-100
0032	7660		-0120	/-50
0033	7730		-0050	/-40
0034	7754		-0024	/-20
0035	7766		-0012	/-10
0036	0000	FRAMEN,	0	/NUMBER OF FRAMES TO DATE
0037	0000	LY,	0	/CURRENT # OF SUBFRAMES IN CURRENT FRAM
0040	0001	CFRAME,	1	/CURRENT FRAME BEING SERVICED
0041	0000	LASTAD,	0	/FIRST USEABLE SF BASE ADDR
0042	0000	OLDADD,	0	
0043	0000	WCOUNT,	0	
0044	0000	WRDMAX,	0	
0045	0000	ALSTBF,	0	
0046	0000	BLSTBF,	0	
0047	0000	ALSTAD,	0	
0050	0000	BLSTAD,	0	
0051	0000	WDBITS,	0	
0052	0000	ZVNODD,	0	
0053	0000	POINTE,	0	
0054	0000	FRMNBR,	0	
0055	0000	COUNTR,	0	
0056	0000	TEMP,	0	
0057	0000		0	
0060	0374	ZREPET,	REPEAT	
0061	0400	ZBPNSR,	BPNSR	
0062	2460	ZLNCHG,	LNCHNG	
0063	0310	ZPRINT,	PRINT	
0064	0200	ZMESAG,	MESAGE	
0065	2512	ZMASF,	MASF	

05398

0066	0366	ZCRLF,	CRLF	
0067	1211	ZBCDPT,	BCDPT	
0070	0366	ZLF,	LF	
0071	0245	ZLSTLC,	LSTLC	
0072	0600	ZPACK,	PACK	
0073	1200	ZCHECK,	CHECK	
0074	1217	ZQUEST,	QUEST	
0075	0500	ZDUMP,	DUMP	
0076	0333	ZBCD,	BCD	
0077	0071	ZLSTSP,	LSTSP	
0100	0472	ZPUN,	PUN	
0101	0477	ZREAD,	READ	
0102	2200	ZOUTPT,	OUTPUT	/THIS SHOULD CONTAIN SIN PNCH,
0103	1177	DECNOL,	TAD L12	
0104	1176	DECNOL,	TAD L-272	
0105	0001	CFBASE,	1	
0106	7740	M40,	-40	
0107	0340	C340,	340	
0110	0245	C245,	245	
0111	0000	MSRGHT,	0	/TEMPORARY STORAGE
0112	0000	CKSM,	0	
0113	7775	N3,	-3	
0114	0000	1A,	0	
0115	0000	FA,	0	
0116	0200	C200,	200	
0117	0177	SL7,	177	
0120	1103	NEXT,	NATLAD	
0121	2534	ZSPACE,	SPACER	
				* 200
0200	0000	MESSAGE,	0	
0201	7240		CLA CMA	/SET C(CAC)=-1
0202	1200		TAD MESSAGE	/ADD LOCATION
0203	3017		DCA 17	/AUTO-INDEX REGISTER
0204	1417		TAD 1 17	/FETCH FIRST WORD
0205	3111		DCA MSRGHT	/SAVE IT
0206	1111		TAD MSRGHT	
0207	7012		RTN	
0210	7012		RTN	/ROTATE 6 BITS RIGHT
0211	7012		RTN	
0212	4216		JMS TYPECH	/TYPE IT
0213	1111		TAD MSRGHT	/GET DATA AGAIN
0214	4216		JMS TYPECH	/TYPE RIGHT HALF
0215	5204		JMP MESSAGE+4	/CONTINUE
0216	0000	TYPECH,	0	/TYPE CHARACTER IN C(CAC)6-11
0217	0175		AND L77J	
0223	7450		SNA	/IS IT END OF MESSAGE?
0221	5417		JMP 1 17	/YES: EXIT
0222	1106		TAD M40	/SUBTRACT 40
0223	7500		SMA	/<40?
0224	5227		JMP .+3	/NO
0225	1107		TAD C340	/YES: ADD 300
0226	5242		JMP MTP	/TO CODES <40
0227	1174		TAD L-3J	/SUBTRACT 3
0230	7440		SZA	/IS IT ZERO?
0231	5234		JMP .+3	/NO
0232	1173		TAD L212J	/YES: CODE 43 IS
0233	5242		JMP MTP	/LINE-FEED (212)
0234	1172		TAD L-2	/SUBTRACT 2
0235	7440		SZA	/IS IT ZERO?
0236	5241		JMP .+3	/NO
0237	1171		TAD L215J	/YES: CODE 45 IS

05399

0240	5242	JMP MTP	/CARRIAGE-RETURN (215)
0241	1110	TAD C245	/ADD 200 TO OTHERS >40
0242	4500	JMS I ZPUN	
0243	7200	CLA	/SET: CLEAR C(AC)
0244	5616	JMP I TYPECH	/RETURN
0245	0000	0	/SUBROUTINE TO SET UP LIST STORAGE
0246	1056	TAD TEMP	/READ # OF WORDS IN FRAME
0247	3410	DCA I BASE	/STORE IN BASE ADDR LIST (AUTO IX)
0250	3441	DCA I LASTAD	/CLEAR POINTER
0251	1041	TAD LASTAD	/BASE ADDRESS OF FRAME LIST
0252	3410	DCA I BASE	/THIS IS NOW FIRST COLUMN OF BASE LIST
0253	7100	CLL	
0254	1056	TAD TEMP	
0255	7010	RAR	
0256	1056	TAD TEMP	
0257	7430	SZL	/CHECK FOR ODD # OF WDS IN FRAME
0260	7001	IAC	
0261	1170	TAD L2	
0262	3057	DCA TEMP+1	
0263	1041	TAD LASTAD	
0264	3042	DCA OLDADD	
0265	1057	TAD TEMP+1	/UPDATE NEXT FRAME LIST BASE ADDR
0266	1041	TAD LASTAD	
0267	3041	DCA LASTAD	
0270	2036	ISZ FRAMEN	
0271	1170	TAD L2	
0272	1042	TAD OLDADD	
0273	3410	DCA I BASE	/STORE IT
0274	1170	TAD L2	
0275	1042	TAD OLDADD	
0276	1056	TAD TEMP	
0277	3410	DCA I BASE	/STORE IT
0300	1167	TAD L-1373	
0301	1010	TAD BASE	/SHOULD BE LOWER
0302	7710	SPA CLA	/ALLOW SLOP OF 4 WORDS
0303	5645	JMP I LSTLOC	
0304	4464	JMS I ZMESAG	
0305	0622	TEXT "FR	
0306	7600	>"	
0307	5645	JMP I LSTLOC	
0310	0000	0	
0311	1056	TAD TEMP	
0312	7006	RTL	
0313	7006	RTL	
0314	3056	DCA TEMP	
0315	1056	TAD TEMP	
0316	0166	AND L7	
0317	1165	TAD L260	
0320	4500	JMS I ZPUN	
0321	7200	CLA	
0322	2332	ISZ M4	
0323	5327	JMP PTLOOP	
0324	1164	TAD L-4	
0325	3332	DCA M4	
0326	5710	JMP I PRINT	
0327	1056	TAD TEMP	
0330	7004	RAL	
0331	5313	JMP PRINT+3	
0332	7774	M4,	-4
		/BINARY TO BCD CONVERSION 3/6/65-AB-DEC	

05400

0333	0000	BCD,	0	
0334	3023		DCA INPUT	/STORE BINARY
0335	1022		TAD CONTROL	/SET UP TABLE
0336	3343		DCA POINTNR	/POINTERS
0337	7100		CLL	
0340	1025		TAD COUNT	
0341	3024		DCA NUMBER	/WILL PUT IT IN LINK
0342	1023		TAD INPUT	
0343	1026	POINTNR,	TAD TABLE	/OR TABLE+1, TABLE+2, ETC.
0344	7430		SZL	/IF C(L)=1, INPUT>-TABLE
0345	3023		DCA INPUT	/IF SO: INPUT=INPUT+TABLE
0346	7200		CLA	
0347	1024		TAD NUMBER	
0350	7004		RAL	/PUT THIS BIT IN ANSWER
0351	2343		ISZ POINTNR	/UPDATE TABLE POINTER
0352	7420		SNL	/IF LINK=1, ALL DONE
0353	5341		JMP POINTNR-2	
0354	7106		CLL RAL	/CONVERTED 2 BCD
0355	7006		RCL	/CHARACTERS
0356	1023		TAD INPUT	/SHIFT LEFT AND ADD
0357	5733		JMP 1 BCD	/THE THIRD
0360	0000	CRLF,	0	
0361	7200		CLA	
0362	1171		TAD 1215	
0363	4500		JMS 1 ZPUN	
0364	4366		JMS LF	
0365	5760		JMP 1 CRLF	
0366	0000	LF,	0	
0367	7200		CLA	
0370	1173		TAD 1212	
0371	4500		JMS 1 ZPUN	
0372	7300		CLA CLL	
0373	5766		JMP 1 LF	
0374	2053	REPEAT,	ISZ POINTE	
0375	5477		JMP 1 ZLSTSF	
				* 400
0400	0000	BPUNSR,0		
0401	7300		CLA CLL	
0402	4240		JMS PLOT /GO PUNCH LEADER CODES	
0403	1600		TAD 1 BPUNSR	
0404	3114		DCA IA	
0405	2200		ISZ BPUNSR	
0406	1600		TAD 1 BPUNSR	
0407	7000		NOP	
0410	3115		DCA FA	
0411	2200		ISZ BPUNSR	
0412	1114		TAD IA	
0413	7120		STL /TO PUNCH IA AS ORIGIN	
0414	4252	PUNL,	JMS BINP /GO PUNCH WORD AS TWO LINES OF TAPE	
0415	1114		TAD IA	
0416	7041		CIA	
0417	1115		TAD FA /AC=FA-IA	
0420	7650		SNA CLA /WAS IT LAST WORD OF BLOCK?	
0421	5226		JMP .+5 /IT WAS THE LAST WORD	
0422	1514		TAD 1 IA /GET WORD TO PUNCH	
0423	7100		CLL /NOT AN ORIGIN	
0424	2114		ISZ IA /JUST INDEX IA	
0425	5214		JMP PUNL	
0426	2113		ISZ NB /IS THERE ANOTHER BLOCK?	
0427	5203		JMP BPUNSR+3 /HANDLE NEXT BLOCK	

05401

0430	1112		TAD CKSM
0431	7100		CLL
0432	4252		JMS BINP /GO PUNCH CHECK SUM
0433	4240		JMS PLOT /GO PUNCH TRAILER CODES
0434	1174		TAD L-31 /NUMBER OF BLOCKS
0435	3113		DCA NB
0436	3112		DCA CKSM
0437	5600		JMP I BPUNSK
0438	0000	PLOT,	0
0441	7300		CLA CLL
0442	1163		TAD L-263 /TO PUNCH 263 OCTAL LEADER /TRAILER CODES
0443	3056		DCA TEMP
0444	1116		TAD C200 /LEADER CODE
0445	4272		JMS PUN /PUNCH C (AC)
0446	2056		ISZ TEMP /ANOTHER L-T CODE OR NOT?
0447	5245		JMP .-2 /GO PUNCH ANOTHER
0450	7200		CLA
0451	5640		JMP I PLOT /EXIT
0452	0000	BINP,	0
0453	3057		DCA TEMP+1
0454	1057		TAD TEMP+1
0455	7312		RTR
0456	7012		RTR
0457	7012		RTR
0460	0117		AND SL7 /FIRST TWO OCTAL DIGITS IN AC 5-11
0461	4272		JMS PUN /PUNCH C (AC)
0462	1112		TAD CKSM
0463	3112		DCA CKSM
0464	1057		TAD TEMP+1
0465	0175		AND L77J /LAST TWO OCTAL DIGITS IN AC 6-11
0466	4272		JMS PUN /PUNCH C (AC)
0467	1112		TAD CKSM
0470	3112		DCA CKSM
0471	5652		JMP I BINP /EXIT
0472	0000	PUN,	0 /ROUTINE TO PUNCH C (AC)
0473	6046		TLS /AND EXIT WITH C (AC)
0474	6041		TSF /UNALTERED
0475	5274		JMP .-1 /PUNCH IT
0476	5672		JMP I PUN /EXIT
0477	0000	READ,	0
0500	6031		KSF /READ THE CHARACTER
0501	5300		JMP .-1
0502	6036		KRB
0503	5677		JMP I READ
0504	0000	DUMP,	0
0505	7200		CLA
0506	1164		TAD L-4
0507	3057		DCA TEMP+1
0510	1162		TAD L-10
0511	3357		DCA WORD
0512	1704		TAD I DUMP
0513	0162		AND L-10
0514	3015		DCA 15
0515	7040		CMA
0516	1015		TAD 15
0517	3015		DCA 15
0520	2304		ISZ DUMP
0521	1704		TAD I DUMP
0522	0162		AND L-10

05402

0523	1166		TAD L7
0524	7041		CIA
0525	1015		TAD 15
0526	3360		DCA FINLOC
0527	2304		ISZ DUMP
0530	4466	NEWLIN,	JMS I ZCRLF
0531	1015		TAD 15
0532	7001		IAC
0533	3056		DCA TEMP
0534	4463		JMS I ZPRINT
0535	1161		TAD L240
0536	4500		JMS I ZPUN
0537	7200	NEWORD,	CLA
0540	1161		TAD L240
0541	4500		JMS I ZPUN
0542	7200		CLA
0543	1415		TAD 1 15
0544	3056		DCA TEMP
0545	4463		JMS I ZPRINT
0546	2357		ISZ WORD
0547	5337		JMP NEWORD
0550	7200		CLA
0551	1162		TAD L-10
0552	3357		DCA WORD
0553	2360		ISZ FINLOC
0554	5330		JMP NEWLIN
0555	4466		JMS I ZCRLF
0556	5704		JMP I DUMP
0557	3000	WORD,	J
0560	0000	FINLOC,	J
0561	1172	DITTOP,	TAD L-2 /AMPERSAND
0562	7450		SNA
0563	5366		JMP DITTOP
0564	1160		TAD L-52
0565	5771		JMP I DITTOP
0566	3046	DITTOP,	DCA BLSTBF
0567	2053		ISZ POINTE
0570	5771		JMP I .+1
0571	0623	DITTOP,	623
* 600			
0600	0000	PACK,	0
			/SUBROUTINE TO READ, CHECK,
			/PACK DECIMAL NUMBERS UP TO 4095.
			/RESULT IS RETURNED IN AC.
			/ A NON NUMBER GIVES A DIAGNOSTIC ? ?
			/A NUMBER EXCEEDING 4095 WILL GIVE
			/A DIAG ?>?, AND RESTART.
			/COMMA IS TERMINAL CHARACTER.
			/ERROR RECOVERY IS BY TYPING ?RUBOUT?.
			/WHICH RE-INITIALIZES AND RESTARTS.
			/IF FIFTH CHAR IS NOT A COMMA,
			/A "?" DIAG WILL BE GIVEN.
0601	4501		JMS I ZREAD
0602	4500		JMS I ZPUN
0603	1157		TAD L-242 /REPEAT (QUOTE)
0604	7450		SNA
0605	5460		JMP I ZREPET
0606	1156		TAD L-15 /SLASH
0607	7450		SNA
0610	5600		JMP I PACK
0611	1155		TAD L-36 /M
0612	7450		SNA

05403

0613	5465	JMP I ZMASF
0614	1154	TAD L15 / AT SIGN
0615	7450	SNA
0616	5462	JMP I ZLNCHG
0617	1153	TAD L+34 /S
0620	7450	SNA
0621	5370	JMP OCTSEL
0622	5773	JMP I DITTO
0623	7450	SNA
0624	5502	JMP I ZOUTPT
0625	1152	TAD L320
0626	7041	CIA
0627	1151	TAD L"
0630	7450	SNA /CHECK FOR COMMA
0631	5275	JMP PCOMMA
0632	3364	DCA PKTEMP /CHECK FOR "," = FIFTH CHAR
0633	7001	IAC
0634	1365	TAD PCKCNT
0635	7450	SNA
0636	5323	JMP PQUEST /TYPE A "?" FOR A DIAG IF NO ","
0637	7200	CLA
0640	1364	TAD PKTEMP
0641	1150	TAD L123 /CHECK FOR RUBOUT
0642	7450	SNA
0643	5324	JMP PQUEST+1 /RE-START
0644	1147	TAD L-377
0645	7041	CIA
0646	4473	JMS I ZCHECK /LOOK FOR DECIMAL OR OCTAL #
0647	7510	SPA
0650	5301	JMP PNUMBR /# DIAGNOSTIC ROUTINE
0651	3364	DCA PKTEMP
0652	1363	TAD PACKWS /CHECK FOR OVERFLOW, CONVT TO DEC OR OCT
0653	7100	CLL
0654	7004	RAL
0655	7430	SZL
0656	5316	JMP OVRFLO
0657	7004	RAL
0660	7430	SZL
0661	5316	JMP OVRFLO
0662	1363	OCTNOP, TAD PACKWS /NOP, IF OCTPCK
0663	7004	RAL
0664	7430	SZL
0665	5316	JMP OVRFLO
0666	1364	TAD PKTEMP
0667	7430	SZL
0670	5316	JMP OVRFLO
0671	3363	DCA PACKWS /END OF CONVSN AND O-F ROUTINE
0672	2365	ISZ PCKCNT
0673	5201	JMP PACK + 1
0674	5306	JMP OUT
0675	1365	PCOMMA, TAD PCKCNT
0676	1146	TAD L5
0677	7440	SZA
0700	5306	JMP OUT
0701	7200	PNUMBR, CLA /# DIAG ROUTINE
0702	1145	TAD L"#
0703	4500	JMS I ZPUN
0704	7200	CLA

05404

0705	5401		JMP PACK + 1
0706	7200	OUT,	CLA /FINALIZATION ROUTINE
0707	1144		TAD L-5
0710	3365		DCA PCKCNT
0711	1363		TAD PACKWS
0712	3364		DCA PKTEMP
0713	3363		DCA PACKWS
0714	1364		TAD PKTEMP
0715	5600	OCTDEC,	JMP 1 PACK /JMP OCTOUT, IF OCTPCK
0716	7300	OVRFLO,	CLA CLL
0717	1143		TAD L">
0720	4500		JMS 1 ZPUN
0721	7200		CLA
0722	7410		SKP
0723	4474	REQUEST,	JMS 1 ZQUEST
0724	1144		TAD L-5 /RE-INITIALIZE
0725	3365		DCA PCKCNT
0726	3363		DCA PACKWS
0727	5201		JMP PACK + 1
0730	0000	OCTPCK,	0 /SAME AS FORPACK, BUT FOR OCTAL #'S
0731	7330		CLA CLL
0732	1356		TAD OCTNOH
0733	3766		DCA 1 LHIGH
0734	1357		TAD OCTNOL
0735	3767		DCA 1 LLOW
0736	1142		TAD L7000
0737	3262		DCA OCTNOP
0740	1360		TAD OCTJMP
0741	3315		DCA OCTDEC
0742	5201		JMP PACK + 1
0743	3364	OCTOUT,	DCA PKTEMP
0744	1104		TAD DECN0H
0745	3766		DCA 1 LHIGH
0746	1103		TAD DECNOL
0747	3767		DCA 1 LLOW
0750	1361		TAD PKTPOP
0751	3262		DCA OCTNOP
0752	1362		TAD DECJMP
0753	3315		DCA OCTDEC
0754	1364		TAD PKTEMP
0755	5730		JMP 1 OCTPCK
0756	1141	OCTNOH,	TAD L-270
0757	1140	OCTNOL,	TAD L10
0760	5343	OCTJMP,	JMP OCTOUT
0761	1363	PKTPOP,	TAD PACKWS
0762	5600	DECJMP,	JMP 1 PACK
0763	0000	PACKWS,	0
0764	0000	PKTEMP,	0
0765	7773	PCKCNT,	-5
0766	1201	LHIGH,	HIGH
0767	1204	LLOW,	LOW
0770	1200	OCTSEL,	TAD PACK
0771	3330		DCA OCTPCK
0772	5331		JMP OCTPCK+1
0773	0561	DITTOL,	DITTOP
			* 1000
1000	7100	BUILDR,	CLL
1001	6042		TCF
1002	6032		KCC
1003	1001		TAD BASBAS

1004	3010		DCA BASE	
1005	1002		TAD AFBASE	
1006	3041		DCA LASTAD	
1007	1105		TAD CFBASE	
1010	3040		DCA CFRAME	
1011	1006		TAD CNIBAS	
1012	3011		DCA CNTADD	
1013	3036		DCA FRAMEN	
1014	3043		DCA WCOUNT	
1015	3037		DCA SFTALY	
1016	4466		JMS I ZCRLF	
1017	4464		JMS I ZMESAG	
1020	1506		TEXT SMF	
1021	0240	B		
1022	7540	=		
1023	0000	\$		
1024	4472		JMS I ZPACK	
1025	3020		DCA MONBIT	
1026	4466		JMS I ZCRLF	
1027	4464		JMS I ZMESAG	
1030	1506		TEXT SMF	
1031	2740	W		
1032	7540	=		
1033	0000	\$		
1034	4472		JMS I ZPACK	
1035	3056		DCA TEMP	
1036	4471		JMS I ZLSTLC	
1037	4405	NEWFRM,	JMS I ZLFLF	
1040	4464		JMS I ZMESAG	
1041	0622		TEXT SFR	
1042	4000	\$		
1043	1040		TAD CFRAME	
1044	4407		JMS I ZDEOPT	
1045	4405		JMS I ZLFLF	
1046	7040		CMA	
1047	1040		TAD CFRAME	
1050	7006		RTL	
1051	0164		AND L7774	
1052	7001		IAC	
1053	1001		TAD BASBAS	
1054	3056		DCA TEMP	/NOW CONTAINS LOC OF C-FR BE
1055	1456		TAD I TEMP	/FROM WORD QTY
1056	3044		DCA WRDMAX	/MAX WORDS IN THIS FRAME
1057	1044		TAD WRDMAX	
1060	7041		CIA	
1061	3043		DCA WCOUNT	/WORD POINTER
1062	1170		TAD L2	
1063	1056		TAD TEMP	/CONSTRUCT ADDR
1064	3056		DCA TEMP	
1065	1456		TAD I TEMP	/PICK UP ADDR OF FIRST WD IN
1066	3050		DCA BLSTAD /	DATA LIST
1067	3450		DCA I BLSTAD	
1070	7001		IAC	/ADD ONE
1071	1056		TAD TEMP	/RE-CONSTRUCT
1072	3056		DCA TEMP	
1073	1456		TAD I TEMP	/PICK UP ADDR OF FIRST WD IN
1074	3047		DCA ALSTAD	/DESCRIPTOR LIST
1075	7040		CMA	
1076	1050		TAD BLSTAD	/CONSTRUCT MAX ADDR

05406

1077	3056		DCA TEMP	
1100	1043		TAD WCOUNT	
1101	7040		CMA	/MAX = ONE LESS THAN NO. OF WDS
1102	3456		DCA 1 TEMP	/STORE MAX
1103	4464	NXTWRD,	JMS 1 ZMESAG	
1104	2704		TEXT SWD	
1105	4000	\$		
1106	1044		TAD WRDMAX	/ASSUME WCOUNT IS TWO'S COMP, ONE LES
1107	1043		TAD WCOUNT	
1110	7001		IAC	
1111	3053		DCA POINTE	/SAVE POINTER FOR C.W. IN C.F.
1112	1053		TAD POINTE	
1113	4407		JMS 1 ZDECP	
1114	5715		JMP 1 P8	
1115	1600	P8,	PAGES	
1116	0000	DECPRT,	0	
1117	4476		JMS 1 ZBCD	/CONVERT CURRENT FR # TO BCD
1120	3056		DCA TEMP	
1121	1056		TAD TEMP	
1122	7004		RAL	
1123	7006		RTL	
1124	7006		RTL	
1125	4467		JMS 1 ZBCDPT	
1126	1056		TAD TEMP	
1127	7012		RTR	
1130	7012		RTR	
1131	4467		JMS 1 ZBCDPT	
1132	1056		TAD TEMP	
1133	4467		JMS 1 ZBCDPT	
1134	5716		JMP 1 DECPRT	
1135	0000	LFLFLF,	0	
1136	4466		JMS 1 ZCRLF	
1137	4470		JMS 1 ZLF	
1140	4470		JMS 1 ZLF	
1141	4470		JMS 1 ZLF	
1142	5735		JMP 1 LFLFLF	
				* 1200
1200	0000	CHECK,	0	/SUBROUTINE TO VERIFY THAT
				/AN ASCII CHAR IS A DECIMAL #.
				/IF A DEC. #, THE OCTAL FORM IS
				/RETURNED IN THE AC.
				/IF NOT A DECIMAL #, MINUS ONE IS
				/RETURNED IN THE AC.
				/HAS OCTAL VERSION ALSO.
				/(SEE OCTPCK)
1201	1176	HIGH,	TAD 1-272	/HIGH ERROR CHECK
1202	7500		SMA	
1203	5207		JMP 1+4	
1204	1177	LOW,	TAD 112	/LOW ERROR CHECK
1205	7500		SMA	
1206	5600		JMP 1 CHECK	
1207	7240		CLA CMA	/RETURN A MINUS ONE
1210	5600		JMP 1 CHECK	
1211	0000	BCDPRT,	0	
1212	0137		AND 117	
1213	1165		TAD 1260	
1214	4500		JMS 1 ZPUN	
1215	7300		CLA CLL	
1216	5611		JMP 1 BCDPRT	
1217	0000	QUEST,	0	
1220	7200		CLA	

05407

1221	1136		TAD L'?
1222	4500		JMS I ZPUN
1223	7200		CLA
1224	5617		JMP I QUEST
1225	0100	NAT,	B /BASE ADDRESS LIST POSITION * 1600
1600	1161	PAGE8,	TAD L240
1601	4500		JMS I ZPUN
1602	7200		CLA
1603	3046		DCA BLSTBF /CLEAR B LIST BUFFER (STUFF LATER)
1604	3045		DCA ALSTBF /CLR A LST BUFFER (E/O TEST STUFF LATE
1605	4464		JMS I ZMESAG
1606	2723		TEXT "WS
1607	4075	=	
1610	4000	"	
1611	4472		JMS I ZPACK
1612	7450		SNA / ZERO => WD SIZE = 10 BITS
1613	5234		JMP TENBIT
1614	3051		DCA WDBITS
1615	1051		TAD WDBITS
1616	1135		TAD L-7
1617	7550		SPA SNA /JMP 7B IF # OF BITS =< 7
1620	5223		JMP SEVENB
1621	3045		DCA ALSTBF /DEPOSIT WRD SIZE FLAGS IN "A" LST
1622	5236		JMP SUBMES
1623	7201	SEVENB,	CLA IAC
1624	3764		DCA I LBTFLG
1625	1051		TAD WDBITS
1626	7012		RTR /LEFT JUSTIFY TO BIT 2+9
1627	7012		RTR
1630	7012		RTR
1631	0134		AND L1600
1632	3046		DCA BLSTBF /PUT # OF BITS INTO DATA WD
1633	5236		JMP SUBMES
1634	1133	TENBIT,	TAD L3
1635	3045		DCA ALSTBF /DEPOSIT WD SIZE FLAGS
1636	4464	SUBMES,	JMS I ZMESAG
1637	2306		TEXT \$SF
1640	1440	L	
1641	7540	=	
1642	0000	\$	
1643	4472		JMS I ZPACK
1644	7450		SNA /JUMP TO SPL CASES IF NO SF HERE
1645	5262		JMP SPLCSS
1646	3056		DCA TEMP
1647	4471		JMS I ZLSTLC
1650	2037		ISZ SFTALY
1651	4464		JMS I ZMESAG
1652	2306		TEXT "SF
1653	4075	=	
1654	4000	"	
1655	1036		TAD FRAMEN
1656	4407		JMS I ZDECPY
1657	1042		TAD OLDADD
1660	3046		DCA BLSTBF
1661	5477		JMP I ZLSTSF /UPDATE "A" + "B" LISTS
1662	4464	SPLCSS,	JMS I ZMESAG
1663	2320		TEXT "SP
1664	1440	L	
1665	7540	=	
1666	0000	"	
1667	7200		CLA

1670	1361		TAD SPCLHI
1671	3762		DCA I LLHIGH
1672	1360		TAD SPCLOW
1673	3763		DCA I LLLON
1674	4472		JMS I ZPACK
1675	3056		DCA TEMP
1676	1104		TAD DECNH
1677	3762		DCA I LLHIGH
1700	1103		TAD DECNOL
1701	3763		DCA I LLLON
1702	1056		TAD TEMP
1703	7100		CLL /STORE SPL CASE IN "A" LIST
1704	0133		AND 13
1705	7006		RTL
1706	7006		RTL
1707	1045		TAD ALSTBF
1710	3045		DCA ALSTBF
1711	4464		JMS I ZMESAG
1712	2431		TEXT "TY
1713	2005	PE	
1714	4075	=	
1715	4000	"	
1716	4501	RDTYPE,	JMS I ZREAD
1717	4500		JMS I ZPUN
1720	3056		DCA TEMP
1721	1161		TAD 1240
1722	4500		JMS I ZPUN
1723	7200		CLA
1724	1056		TAD TEMP
1725	1132		TAD 1-306 /LOOK FOR "F"
1726	7450		SNA
1727	5343		JMP FIXED
1730	1131		TAD 1-14 /LOOK FOR "R"
1731	7450		SNA
1732	5755		JMP 1 RNDM
1733	1130		TAD 111 /LOOK FOR "I"
1734	7450		SNA
1735	5756		JMP 1 INCR8
1736	1146		TAD 15 /LOOK FOR "D"
1737	7450		SNA
1740	5757		JMP 1 DECR8
1741	4474		JMS I ZQUEST
1742	5316		JMP RDTYPE
1743	4464	FIXED,	JMS I ZMESAG
1744	0401		TEXT "DA
1745	2401	TA	
1746	4075	=	
1747	4000	"	
1750	4754		JMS I OCTALP
1751	1046		TAD BLSTBF
1752	3046		DCA BLSTBF
1753	5477		JMP 1 ZLSTSF
1754	0730	OCTALP,	OCTPCK
1755	2000	RNDM,	RANDOM
1756	2006	INCR8,	INCR
1757	2056	DECR8,	DECR
1760	1133	SPCLOW,	TAD 1 3
1761	1163	SPCLHI,	TAD 1-263
1762	1201	LLHIGH,	HIGH
1763	1204	LLLOW,	LOW
1764	2067	LBTFLG,	BITFLG

05409

* 2000

2000	1127	RANDOM,	TAD L4
2001	1045		TAD ALSTBF
2002	3045		DCA ALSTBF
2003	1003		TAD RANDAD
2004	3046		DCA BLSTBF
2005	5271		JMP LSTSTF
2006	1140	INCR,	TAD L10
2007	1045		TAD ALSTBF
2010	3045		DCA ALSTBF
2011	4464		JMS I ZMESAG
2012	1120		TEXT "IP
2013	4075	=	
2014	4000	"	
2015	4472		JMS I ZPACK
2016	7000		NOP
2017	7450		SNA
2020	5235		JMP ZEROIP
2021	3056	NZIP,	DCA TEMP
2022	1056		TAD TEMP
2023	1126		TAD L4000
2024	4240		JMS RPTTEXT
2025	1056		TAD TEMP
2026	3411		DCA I CNTADD
2027	1172		TAD L-2
2030	1011		TAD CNTADD
2031	3046		DCA BLSTBF
2032	1142		TAD L7000
2033	3216		DCA NZIP-3
2034	5271		JMP LSTSTF
2035	4240	ZEROIP,	JMS RPTTEXT
2036	7040		CMA
2037	5230		JMP .-7
2040	0000	RPTTEXT,	0
2041	3273		DCA CNTEMP
2042	1267		TAD BITFLG
2043	7640		SZA CLA
2044	5263		JMP SVNCNT
2045	1270		TAD CNTEMP
2046	3411	CNTPNT,	DCA I CNTADD
2047	4464		JMS I ZMESAG
2050	2220		TEXT "RP
2051	4075	=	
2052	4000	"	
2053	4472		JMS I ZPACK
2054	3411		DCA I CNTADD
2055	5640		JMP I RPTTEXT
2056	1262	DECR,	TAD DCRJMP
2057	3216		DCA NZIP-3
2060	1125		TAD L14
2061	5207		JMP INCR+1
2062	5221	DCRJMP,	JMP NZIP
2063	3267	SVNCNT,	DCA BITFLG /CLEAR FLAG
2064	1046		TAD BLSTBF
2065	1270		TAD CNTEMP
2066	5246		JMP CNTPNT
2067	0000	BITFLG,	0
2070	0000	CNTEMP,	0
2071	4466	LSTSTF,	JMS I ZCRLF
2072	1053		TAD POINTE

05410

2073	7510		RAK	/CHECK EVEN-ODD
2074	7204		GLK	
2075	3052		DCA EVNODD	
2076	3267		DCA BITFLG	/CLEAR IT FOR NEXT DATA WORD
2077	1052		TAD EVNODD	
2100	7650		SNA CLA	
2101	5310		JMP EVNCNT	
2102	1045		TAD ALSTBF	
2103	7006		RTL	
2104	7006		RTL	
2105	7006		RTL	
2106	0124		AND I7700	
2107	5312		JMP .+3	
2110	1045	EVNCNT,	TAD ALSTBF	
2111	1447		TAD I ALSTAD	
2112	3447		DCA I ALSTAD	
2113	1046		TAD BLSTBF	
2114	3450		DCA I BLSTAD	
2115	2043		ISZ WCOUNT	
2116	5341		JMP UPDATE /UPDATE LIST ADDRESSES	
2117	7040		CMA	
2120	1040		TAD CFRAME /CURRENT FRAME #	
2121	7006		RTL /MULTIPLY BY 4	
2122	0164		AND I7774 /GET RID OF JUNK	
2123	1133		TAD I3	
2124	1001		TAD BASBAS /CONSTRUCT ADDR	
2125	3056		DCA TEMP	
2126	1037		TAD SFTALY /STORE IN BASE ADDR LIST	
2127	3456		DCA I TEMP	
2130	3037		DCA SFTALY	
2131	2040		ISZ CFRAME	
2132	1040		TAD CFRAME	
2133	7041		CIA	
2134	1036		TAD FRAMEN	
2135	7510		SPA	
2136	5502		JMP I ZOUTPT	
2137	7200		CLA	
2140	5753		JMP I NEW /START BUILDING A NEW FRAME	
2141	2050	UPDATE,	ISZ BLSTAD	
2142	7300		CLA CLL	
2143	3450		DCA I BLSTAD	
2144	1052		TAD EVNODD /CHECK E-O	
2145	7440		SZA	
2146	5351		JMP .+3	
2147	2047		ISZ ALSTAD	
2150	3447		DCA I ALSTAD	
2151	4472	CHANGE,	JMS I ZPACK	
2152	5520		JMP I NEXT	
2153	1037	NEW,	NEWFRM	
				* 2200
2200	4466	OUTPUT,	JMS I ZCRLF	
2201	4464		JMS I ZMESAG	
2202	2016		TEXT SPN	
2203	0310	CH		
2204	0000	S		
2205	4466		JMS I ZCRLF	
2206	7402		HLT	
2207	7200		CLA	
2210	1123		TAD I20	
2211	3225		DCA P1	

05411

2212	1377		TAD (22)
2213	3226		DCA P1+1
2214	1906		TAD CNTBAS
2215	3227		DCA P1+2
2216	1911		TAD CNTADD
2217	3230		DCA P1+3
2220	1902		TAD NFBASE
2221	3231		DCA P1+4
2222	1041		TAD LASTAD
2223	3232		DCA P1+5
2224	4461		JMS I ZBPNSR
2225	0000	P1,	0
2226	0000		0
2227	0000		0
2230	0000		0
2231	0000		0
2232	0000		0
2233	7200		CLA
2234	7000		NOP /REPLACE WITH (JMP MATRIX)
2235	1344		TAD JMPMAT
2236	3234		DCA *-2
2237	1345		TAD LFRAMN /ACTIVE REGISTERS
2240	3225		DCA P1
2241	1346		TAD LPOINT /ACTIVE REGISTERS
2242	3226		DCA P1+1
2243	1001		TAD BASBAS
2244	3227		DCA P1+2
2245	1010		TAD BASE
2246	3230		DCA P1+3
2247	1140		TAD L10
2250	3231		DCA P1+4
2251	1137		TAD L17
2252	3232		DCA P1+5
2253	5224		JMP P1-1
2254	7402	MATRIX,	HLT
2255	1142		TAD L7000
2256	3234		DCA P1+7
2257	4405		JMS I ZLFLF
			/PRINT MATRIX HERE * * * * *
2260	4464		JMS I ZMESAG
2261	0622		TEXT "FR
2262	0115	AM	
2263	0540	E	
2264	2717	NO	
2265	2204	RD	
2266	2350	SC	
2267	7051	8)	
2270	4002	B	
2271	0123	AS	
2272	0540	E	
2273	2325	SU	
2274	0240	B	
2275	0622	FR	
2276	4004	D	
2277	0523	ES	
2300	0340	C	
2301	0201	BA	
2302	2305	SE	
2303	0000	"	
2304	4466		JMS I ZCRLF
2305	7200		CLA
2306	1036		TAD FRAMEN

05412

2307	7041		CIA
2310	3054		DCA FRMBR /USED FOR COUNTER
2311	3012		DCA 12 /USED AS INDEX FOR FR CNT
2312	1001		TAD BASBAS
2313	3013		DCA 13 /MATRIX ELEMENT ADDR
2314	4521	MXLOOP,	JMS I ZSPACE
2315	0001		1
2316	2012		ISZ 12
2317	1012		TAD 12
2320	4407		JMS I ZDECT
2321	1347		TAD ARGLST /ONE LESS THAN BASE ADDR OF SPACE CONS
2322	3014		DCA 14
2323	1164		TAD L-4
2324	3055		DCA COUNTR
2325	1414		TAD 1 14
2326	3330		DCA ARGMT
2327	4521		JMS I ZSPACE
2330	0000	ARGMT,	0
2331	1413		TAD 1 13
2332	3056		DCA TEMP
2333	4463		JMS I ZPRINT
2334	2055		ISZ COUNTR
2335	5325		JMP ARGMT-3
2336	4466		JMS I ZCRLF
2337	2054		ISZ FRMBR
2340	5314		JMP MXLOOP
2341	4466		JMS I ZCRLF
2342	5743		JMP I .+1
2343	2400		2400
2344	5254	JMPMAT,	JMP MATRIX
2345	0036	LFRAMB,	FRAMB
2346	0053	LPOINT,	POINTE
2347	2347	ARGLST,	ARGLST
2350	0004		4
2351	0003		3
2352	0002		2
2353	0004		4
2377	0022		*2400
2400	4464		JMS I ZMESAG
2401	0316		TEXT SCN
2402	2401	TA	
2403	0404	DD	
2404	0000	\$	
2405	1011		TAD CNTADD
2406	3056		DCA TEMP
2407	4463		JMS I ZPRINT
2410	1011		TAD CNTADD
2411	3223		DCA D1
2412	4466		JMS I ZCRLF
2413	4464		JMS I ZMESAG
2414	0316		TEXT SCN
2415	2422	TR	
2416	2356	S.	
2417	5656	..	
2420	0000	\$	
2421	4475		JMS I ZDUMP
2422	1400		1400
2423	0000	D1,	0
2424	4466		JMS I ZCRLF
2425	4464		JMS I ZMESAG
2426	1401		TEXT \$LA

05413

2427	2324	ST	
2430	4000	S	
2431	7240		CLA CMA
2432	1041		TAD LASTAD
2433	3056		DCA TEMP
2434	4463		JMS I ZPRINT
2435	4466		JMS I ZCRLF
2436	4464		JMS I ZMESAG
2437	0401		TEXT SDA
2440	2401	TA	
2441	5656	..	
2442	5600	..S	
2443	1002		TAD MFBASE
2444	3251		DCA D2
2445	7040		CMA
2446	1041		TAD LASTAD
2447	3252		DCA D3
2450	4475		JMS I ZDUMP
2451	0000	D2,	0
2452	0000	D3,	0
2453	4405		JMS I ZLFLF
2454	7402		HLT
2455	4466		JMS I ZCRLF
2456	5657		JMP I .+1
2457	1000		1000
2460	1472	LNCHNG,	TAD I ZPACK
2461	7040		CMA
2462	1311		TAD LCHANG /IS CALL FROM UPDATE
2463	7650		SNA CLA
2464	5520		JMP I NEXT /NO
2465	7040		CMA /YES
2466	1050		TAD BLSTAD
2467	3050		DCA BLSTAD /RESTORE
2470	3450		DCA I BLSTAD /CLEAR
2471	7040		CMA
2472	1043		TAD WCOUNT
2473	3043		DCA WCOUNT /RESTORE
2474	1052		TAD EVNODD
2475	7450		SNA
2476	5301		JMP .+3 /EVEN WORD
2477	7200		CLA /ODD
2500	3045		DCA ALSTBF /CLEAR
2501	5520		JMP I NEXT
2502	1045		TAD ALSTBF
2503	0124		AND 17700
2504	3045		DCA ALSTBF /RESTORE
2505	7040		CMA
2506	1047		TAD ALSTAD
2507	3047		DCA ALSTAD /RESTORE ADDRESS
2510	5520		JMP I NEXT
2511	2151	LCHANG,	CHANGE
2512	4521	MASF,	JMS I ZSPACE
2513	0001		1
2514	4464		JMS I ZMESAG
2515	2306		TEXT "SF
2516	4075	=	
2517	4000	"	
2520	4472		JMS I ZPACK
2521	3056		DCA TEMP
2522	7040		CMA /SUBTRACT 1

05414

2523	1056	TAD TEMP	
2524	7006	RTL	
2525	0164	AND L7774	
2526	1170	TAD L2	
2527	1001	TAD BASBAS	
2530	3056	DCA TEMP	
2531	1456	TAD I TEMP	
2532	3046	DCA BLSTBF	
2533	5477	JMP I 2LSTSF	
2534	0000	SPACER,	0
2535	1734	TAD I SPACER	
2536	2334	ISZ SPACER	
2537	7041	CIA	
2540	3056	DCA TEMP	
2541	1161	TAD L240	
2542	4000	JMS I 2PUN	
2543	2056	ISZ TEMP	
2544	5342	JMP .-2	
2545	7200	CLA	
2546	5734	JMP I SPACER	
2547	0000	FIRST,	0 /BASE POSITION FOR BIT STREAM
0123	0020		S
0124	7700		
0125	0014		
0126	4000		
0127	0004		
0130	0011		
0131	7764		
0132	7472		
0133	0003		
0134	1600		
0135	7771		
0136	0277		
0137	0017		
0140	0010		
0141	7510		
0142	7000		
0143	0276		
0144	7773		
0145	0243		
0146	0005		
0147	7401		
0150	0123		
0151	0254		
0152	0320		
0153	0034		
0154	0015		
0155	7742		
0156	7763		
0157	7536		
0160	7726		
0161	0240		
0162	7770		
0163	7515		
0164	7774		
0165	0260		
0166	0007		
0167	6405		
0170	0002		
0171	0215		
0172	7776		

05415

0173 0212
0174 7775
0175 0077
0176 7506
0177 0012

ALSTAD 0047
ALSTBF 0045
ARGLST 2347
ARGMNT 2330
BASBAS 0001
BASE 0010
BASEAD 0021
BCD 0333
BCDPRT 1211
BINP 0452
BITFLG 2067
BLSTAD 0050
BLSTBF 0046
BPUNSR 0400
BUILDR 1000
CFBASE 0105
CFRAME 0040
CHANGE 2151
CHECK 1200
CKSM 0112
CNTADD 0011
CNTBAS 0006
CNTEMP 2070
CNTPNT 2046
CONTRL 0022
COUNT 0025
COUNTR 0055
CRLF 0360
C200 0116
C245 0110
C340 0107
DCRJMP 2062
DECJMP 0762
DECNOH 0104
DECNOL 0103
DECPRT 1116
DECR 2056
DECR8 1757
DITTOO 0566

05416

DITTOL 0773
DITTOP 0561
DITTOR 0571
DJMP 0504
D1 2423
D2 2451
D3 2452
EVENONT 2110
EVENODD 0352
FA 0115
FINLOC 0560
FIRST 2547
FIXED 1743
FRAMEN 0036
FRANBR 0054
HIGH 1201
IA 0114
INCR 2006
INCR8 1756
INPUT 0023
JMPMAT 2344
LASTAD 0041
LBTFLG 1764
LCHANG 2511
LF 0366
LFLFLF 1135
LFRAMN 2345
LHIGH 0766
LLHIGH 1762
LLOW 1763
LLOW 0767
LNCHNG 2460
LOW 1204
LPOINT 2346
LSTLOC 0245
LSTSTF 2071
MASF 2512
MAT 1225
MATRIX 2254
MESSAGE 0200
MFBASE 0002
MONBIT 0020
MSRGHT 0111
MTP 0242
MXLOOP 2314
M4 0332
M40 0106
NB 0113
NEW 2153
NEWFRM 1037
NEWLIN 0530
NEWORD 0537
NEXT 0120
NUMBER 0024
NXTWRD 1103
NZIP 2021
OCTALP 1754
OCTDEC 0715
OCTJMP 0760

05417

OCTNOH 0756
 OCTNOL 0757
 OCTNOP 0662
 OCTOUT 0743
 OCTPCK 0730
 OCTSEL 0770
 OLDADD 0042
 OUT 0706
 OUTPUT 2200
 OVRFLO 0716
 PACK 0600
 PACKAS 0763
 PAGES 1600
 PKCNT 0765
 PCOMMA 0675
 PKTEMP 0764
 PKTPOP 0761
 PLOT 0440
 PNUMBER 0701
 POINTE 0053
 POINTR 0343
 PQUEST 0723
 PRINT 0310
 PTLOOP 0327
 PUN 0472
 PUNL 0414
 P1 2225
 P8 1115
 QUEST 1217
 RANDAD 0003
 RANDOM 2000
 RDTYPE 1716
 READ 0477
 REPEAT 0374
 RNDM 1755
 RPTEXT 2040
 SEVENB 1623
 SFTALY 0037
 SL7 0117
 SPACER 2534
 SPCLHI 1761
 SPCLOW 1760
 SPLOSS 1662
 SUBMES 1636
 SVNCNT 2063
 TABLE 0026
 TEMP 0056
 TENBIT 1634
 TYPECH 0216
 UPDATE 2141
 WCOUNT 0043
 WOBITS 0051
 WORD 0557
 WRDMAX 0044
 ZBCD 0076
 ZBCDPT 0067
 ZBPNSR 0061
 ZCHECK 0073
 ZCRLF 0066
 ZDECPT 0007
 ZDUMP 0075
 ZEROIP 2035

ATM-646
 Page 42 of 43

05418

ZLF 0075
ZLFELF 0005
ZLVGHH 0062
ZLSTLC 0071
ZLSTSF 0077
ZMBSP 0065
ZMESAG 0064
ZOUTPT 0102
ZPACK 0072
ZPRINT 0063
ZPUN 0100
ZQUEST 0074
ZREAD 0101
ZREPER 0060
ZSPACE 0121