

## EXEC PARAMETERS 6/21/66 (PARAMS, 31)

SZS=640000  
LAT=762200

GLPHNG=670000  
TYIHNG=700000  
TYOHNG=710000  
RPAHNG=720000  
PPAHNG=730000  
CLKHNG=740000  
EOTHNG=750000  
UNUSED=760000  
CORHNG=770000

BITS=0  
NOCHAR=1  
TOTRT=2  
IOPRT=4  
BILLTT=6  
SUPTD=7  
PROGNM=11  
PRIORITY=15  
DDTSEG=16  
RESTPC=32  
HH16AC=40  
HH16PC=41  
HH16IO=42  
HH16FG=43  
C16AC=44  
C16PC=45  
C16IO=46  
C16FLA=47  
AC=50  
PC=35  
IO=51  
FLAGS=52  
FASTAC=53  
FASTPC=54  
C16TEM=55  
DDTS6=65  
RESTSU=73  
IOPMAX=74  
TISMAX=75  
TTNO=76  
TRAPPC=77  
ERCODE=102  
TTTSU=105

LRG=IOT 10  
ERG=IOT 11  
RRC=IOT 2122            RCC=IOT 3022  
TCB=IOT 4022            SSB=IOT 4122  
RCK=IOT 32  
CKS=IOT 33  
RTB=IOT 35  
DSC=IOT 50  
ASC=IOT 51  
ISB=IOT 52  
CAC=IOT 53  
LSM=IOT 54  
ESM=IOT 55  
CBS=IOT 56  
DIA=IOT 61  
DWC=IOT 62            DRA=IOT 2062  
DCL=IOT 63  
ERM=IOT 65  
RNM=IOT 66  
LEM=IOT 74            EEM=IOT 4074  
RPA=IOT 1 RRB=IOT 30  
PPA=IOT 5

PUNLEN=40  
RDLEN=200  
COR=140000

START

## SEQUENCE BREAK ROUTINES 10/12/66 (SBOUT,41)

0/	OFFSET DCH	
77	JMP DATACH	/1 DATA CHANNEL
13/	JMP READER	/2 READER
23/	JMP CONTRL	/4 CONTROLLERS
27/	JMP LD	/5 SWAPPING DRUM
33/	JMP TTSERV	/6 SCANNER
37/	JMP 1SEC	/7 1 SEC CLOCK
43/	JMP 1MIN	/10 1 MIN CLOCK
53/	JMP PUNCH	/12 PUNCH
63/	JMP SOROBA	/14 TYPEWRITER
67/	JMP IOP	/15 IO PROCESSOR ISB'S
73/	JMP DISPAT	/16 RESTRICT MODE TRAP
77/	JMP SWAP	/17 32 MSEC. CLOCK

200/STAT, TYIHNG /STAT+177 MUST BE LESS THAN 8 BITS (377)  
REPEAT 177, UNUSED

/ TELETYPE TRANSLATION TABLES  
 600/ /TABLES START AT ODD MULTIPLE OF 200

/BIT 0 INDICATES 12 BIT CHAR. ON INPUT  
 /BITS 1-6 ARE INTERNAL CODE OF INPUT CHAR.  
 /BIT 7 INDICATES CHAR. IS CT ALARM CHAR.  
 /BIT 8 INDICATES CHAR. IS NON-CT ALARM

/BITS 10-17 ARE TT OUTPUT CODE

W=402000 /WARNING ALARM CHARACTERS  
 C=002000 /6 BIT CT ALARM CHARACTERS  
 L=000000 /NON ALARM CHARACTERS  
 U=000177 /OUTPUT CHARACTER FOR UNUSED INTERNAL CODES

DEFINE M TYPE, IN, OUT  
 IN "T" 4000+TYPE+OUT  
 TERMINATE M

	/INPUT CHAR	/OUTPUT CHAR
T TTB1,		
M L,00,040	/NULL	/SPACE
M W,01,041	/"A"	/?
M W,02,042	/"B"	/"
M W,03,043	/"C"	/#
M W,04,044	/EOT	/\$
M W,05,045	/WRU	/%
M W,06,046	/RU	/&
M W,07,047	/BELL	/'
M W,10,050	/"H"	/(
M W,11,051	/"TAB	/>)
M W,12,052	/"LF	/*
M W,13,053	/"VT	/+
M W,14,054	/"FORM	/,
M C,76,055	/"CR	/-
M W,16,056	/"N"	/.
M W,17,057	/"O"	//
M W,20,060	/"P"	/0
M W,21,061	/"Q"	/1
M W,22,062	/"TAPE	/2
M W,23,063	/"XOFF	/3
M W,24,064	/"T"	/4
M W,25,065	/"U"	/5
M W,26,066	/"V"	/6
M W,27,067	/"W"	/7
M W,30,070	/"X"	/8
M W,31,071	/"Y"	/9
M W,32,072	/"Z"	/:
3 63073	/"["=AM	/;
M W,34,074	/"BS"	/<
M W,35,075	/""]"	/=
M W,36,076	/"↑"	/>
M W,37,077	/"↔"	/?

M C,00,100	/SPACE	/@
M C,01,101	/! /A	
M L,02,102	/" /B	
M C,03,103	/# /C	
M C,04,104	/\$ /D	
M C,05,105	/% /E	
M C,06,106	/& /F	
M C,07,107	/' /G	
M C,10,110	/(` /H	
M C,11,111	/)` /I	
M C,12,112	/* /J	
M C,13,113	/+ /K	
M C,14,114	/, /L	
M C,15,115	/- /M	
M L,16,116	/• /N	
M C,17,117	/// /O	
M L,20,120	/0 /P	
M L,21,121	/1 /Q	
M L,22,122	/2 /R	
M L,23,123	/3 /S	
M L,24,124	/4 /T	
M L,25,125	/5 /U	
M L,26,126	/6 /V	
M L,27,127	/7 /W	
M L,30,130	/8 /X	
M L,31,131	/9 /Y	
M C,32,132	/: /Z	
M C,33,133	/; /[	
M C,34,175	/< /AM	
M C,35,135	/= /]	
M C,36,015	/> /CRLF	
M C,37, 0	/?	

M C, 40,000	/@ /NULL
M L, 41,001	/A /"A"
M L, 42,002	/B /"B"
M L, 43,003	/C /"C"
M L, 44,004	/D /EOT
M L, 45,005	/E /WRU
M L, 46,006	/F /RU
M L, 47,007	/G /BELL
M L, 50,010	/H /"H"
M L, 51,011	/I /TAB
M L, 52,012	/J /LF
M L, 53,013	/K /VT
M L, 54,014	/L /FORM
M L, 55,015	/M /CR
M L, 56,016	/N /"N"
M L, 57,017	/O /"O"
M L, 60,020	/P /"P"
M L, 61,021	/Q /"Q"
M L, 62,022	/R /TAPE
M L, 63,023	/S /XOFF
M L, 64,024	/T /"T"
M L, 65,025	/U /"U"
M L, 66,026	/V /"V"
M L, 67,027	/W /"W"
M L, 70,030	/X /"X"
M L, 71,031	/Y /"Y"
M L, 72,032	/Z /"Z"
M C, 73,033	/[ /"[
M W, 44,034	/BS /"BS"
M C, 75,035	/] /"]"
M W, 46,036	/↑ /"↑"
M W, 47,037	/← /"←"
REPEAT 4, M L, 0, U	/ /
M L, 0, 134	/ /BS
M L, 0, U /	/
M L, 0, 136	/ /↑
M L, 0, 137	/ /←
REPEAT 20, M L, 0, U	/ /
REPEAT 4, M L, 0, U	/ /
M L, 0, 177	/ /RO
3 63000+U	/EOM /
3 63000+U	/EOM /
7 63000+U	/RO /
TTTBL+200,	EXPUNGE C,L,M,W,U

1000, TTP, /TELETYPE PTRS. (400 WDS)

```
DEFINE TTPTRS NUM,ORG  
REPEAT NUM,[  
101      /LINE NOT OPEN, RNG SET  
LAC 140000+[.-TTP-1]"T"2+0RG          /SERVICE PTR  
LAC 140000+[.-TTP-2]"T"2+0RG          /USER PTR  
0        /PGM WORD (0=DISCONNECTED)  
]  
TERMINATE TTPTRS
```

```
TTPTRS 10, 100      /100-177 IS TT BUFFERS 0-7  
TTPTRS 20, 300      /400-577 IS TT BUFFERS 10-27  
TTPTRS 50, 1100     /1400-2077 IS TT BUFFERS 30-77  
TTP/    105  
TTP+3/   STAT 0
```

/ INITIALIZER

```
2100/  
INIT,      LAC .+2      /FLY THE SCANNER  
DAC 74  
CKS  
RIR 2S  
SPI  
RCC  
ISP 74  
JMP INIT+2  
RTB  
CAC  
CBS  
LAW DSWAP  
DAP 77  
ESM  
ISB 1700  
JMP .  
DSWAP,      LAW SWAP  
DAP 77  
LIO (340000)  /MEMORY PROTECT BITS  
ERM  
IRP [A,,1,2,4,5,6,7,12,14]  
ASC A"T"100  
ENDIRP  
ISB 600      /JUST TO MAKE SURE SCANNER FLYING  
EEM  
JMP I .+1  
DSWAPX:    DCH ESWORG /DAC'ED INTO BY RESTART TAPE
```

DATA CHAN SEQ BRK  
DATACH, LIF"U"SCF"U"CLL  
DIO DATSAV  
EEM  
JMP I DATADR  
DATX, LIO DATSAV  
LFI  
LIO 6  
LAC 4  
JMP I 5  
DATSAV, 0

/ CONTROLLERS SEQ BRK  
CTRL, LIO CONSAV  
LIF"U"LFI  
DIO CONSAV  
EEM  
JMP I CONADR  
CONX, LIO CONSAV  
LIF"U"LFI  
DIO CONSAV  
LAC 20  
LIO 22  
JMP I 21  
CONSAV, 0

/ I/O PROCESSOR LOW PRIORITY SEQ BRK  
IOP, LIF"U"SCF"U"CLL  
DIO IOPSAV  
EEM  
JMP I IOPADR  
IOPX, LIO IOPSAV  
LFI  
LIO 66  
LAC 64  
JMP I 65  
IOPSAV, 0  
/ READER SEQ BRK

DIMENSION RDRBUF(RDLEN)

READER, EEM  
JSP RDR  
LIO 12  
LAC 10  
JMP I 11

R DR, RRB  
RB1, XX / RPA OR DIO RB1  
DAC READEX  
LAC RBI  
SPA  
JMP RB3  
RB4, RIL 9S  
DIO I RBI  
ADD (400000)  
SAD RBE  
SUB (RDLEN)  
DAC RBI  
ADD (1)  
SUB RBE  
RAL 1S  
SMA  
SUB (RDLEN RDLEN)  
RAR 1S  
ADD RBE  
SUB RBO  
RAL 1S  
SPA  
ADD (RDLEN RDLEN)  
SZA I  
JMP RB6  
SAS (RDLEN RDLEN-40)  
JMP RB5  
7 RDRI, LAC RDWHO  
SAD BDSTAT  
LAW FSTAT  
DAP RDVAR  
LAW I 7777 /BUFFER FILLING  
LSM  
AND I RDVAR  
SAS (RPAHNG) /HUNG ?  
JMP RB7 /NO  
LIO (3)  
LAW STAT  
SAD RDWHO  
CLI  
DIO I RDVAR  
ESM  
DZM HOTFLG  
RB5, RCK  
DIO RBK  
JMP I READEX  
RB3, LAC I RBI  
RCL 9S  
LAC RBI  
JMP RB4  
RB6, LIO +1 /BUFFER FULL ON NEXT CHARACTER  
CRB1, DIO RB1  
JMP RB5  
RB7, ESM  
JMP RB5

R DVAR,	DCH
RBI,	COR RDRBUF
RBO,	COR RDRBUF
RBE,	COR RDRBUF+RDLEN
RBK,	0
RDWHO,	-0
RX,	0
READEX,	0

/PUNCH SEQ BRK

DIMENSION PUNBUF(PUNLEN)

PUNCH, EEM  
IDX PRUN  
LAC PSP  
SAD PUP  
JMP PUNX  
DZM PRUN  
LIO I PSP  
PPA  
IDX PSP  
SAD PEND  
SUB (PUNLEN)  
DAC PSP  
ADD (10)  
SAD PUP  
JMP PUNALM  
SUB (PUNLEN)  
SAS PUP /10 CHARS FROM FULL?  
JMP PUNX

PUNALM, LAC PUNWHO  
SAD BDSTAT  
LAW FSTAT  
DAP PVAR  
LAW I 7777  
LSM  
AND I PVAR  
SAS (PPAHNG) /HUNG ?  
JMP PUNCH1 /NO  
LAW 3  
DAC I PVAR  
ESM  
ISB 1700  
DZM HOTFLG

PUNX, LIO 52  
LAC 50  
JMP I 51

PUNCH1, ESM  
JMP PUNX

PUNWHO, -0  
P SP, COR PUNBUF  
P UP, COR PUNBUF  
P END, COR PUNBUF+PUNLEN  
P VAR, DCH .  
P RUN, 1

/ SOROBAN SEQ BRK  
SOROBA, EEM  
LSM  
JSP I SORADR  
SORX, ESM  
LIO 62  
LAC 60  
JMP I 61

/1 SEC CLOCK  
1 SEC, LIO (1) /CHANNEL 7 ROUTINE TO UNHANG  
LAW STAT /USERS WHO ARE CLOCK HUNG  
1 SECL1, DAP 1PTR  
1 SECL2, LAW I 7777 /LOOP TO SEARCH FOR CLOCK-HUNG  
LSM  
1 PTR, AND •  
SAD (CLKHNG)  
JMP 1SECT2 /FOUND ONE  
1 SECT1, ESM  
IDX 1PTR  
SAD (AND FSTAT+1)  
JMP 1SECC  
ADD (WHERE-STAT-AND)  
SAS WTOP  
JMP 1SECL2  
LAW FSTAT  
JMP 1SECL1  
  
1 SECT2, IDX I 1PTR /COUNT TOWARD 7777  
SAS (CLKHNG+I) /IS IT UNHUNG  
JMP 1SECT1  
DIO I 1PTR  
DZM HOTFLG  
JMP 1SECT1  
  
1 SECC, EEM  
LSM  
LAC OCTR  
DZM OCTR  
ESM  
SZA  
JSP I 1SECJA  
1 SECA, LSM  
LAC 1CTR  
DZM 1CTR  
ESM  
SZA  
JSP I 1SECJB  
1 SECB, SZS I 40  
7 TT2, JSP •+1 /NORMAL •+1; ABNORMAL 7TT4. SET ON CH6  
LAC RB1  
SAS RB2  
JMP 7RDRX  
LSM  
RCK  
LAI  
SUB RBK  
ESM  
SPA  
ADD (60000.)  
SUB (100.)  
SPA  
JMP 7RDRX  
LIO •+1  
DIO RB1  
DSC 200  
LAC (DCH 7RDRXX)  
DAC READEX  
JMP 7RDRI  
7 RDRXX, ASC 200  
7 RDRX, LIO 36  
LAC 34  
JMP I 35

0 CTR,  
1 CTR,

0  
0

/1 MINUTE CLOCK  
1MIN5 EEM  
1MIX, JMP I 1MIADR  
LIO 42  
LAW 1000  
DAP TTF /SEE SWNN DISPLAY HACK  
LAC 40  
JMP I 41

START

## TELETYPE SERVICE ROUTINE 8/15/66 (TTSERV, 32)

TTSERV, LIF  
DIO TTFLGS  
TTLLOOP, ASC 600 /PROCESS ALL WAITING CHARS, THEN DEBREAK  
CKS  
RIR 2S  
SPI I  
JMP TTDBRK /SCANNER NOT STOPPED  
RRC  
SZS 40  
JMP TTSS4 /IGNORE HIGH-NUMBERED TT'S  
TTSS4X, SSB  
LAW TTP"T"200000  
SCF"U"AAI  
SAL 2S  
DAP . 1  
TTF, LIO .  
LFI"U"IDA  
DSC 600 /KEEP WAITING BKS FROM PILING UP  
RCC  
SZF I 6  
JMP TTL0 /LINE OPEN  
SZF I 1  
JMP TTIN /INPUT  
DAP TTALD /OUTPUT  
DAP TTASP  
IDA  
DAP TTAUP  
LAW TTTBL /ODD INTEGRAL MULTIPLE OF 200  
IAI"U"SZL"U"CLL  
JMP TTASP /DON'T TEST FOR NULL OR CR  
SZF 3  
JMP TTINT /BEING INTERRUPTED  
SAD (TTTBL  
JMP TT1NUL /FIRST NULL  
SAD (TTTBL"U"215  
JMP TTACR /CR. SEND A LF.  
TTASP, LAC .  
TTAUP, SAD .  
JMP TTEMPTY /GO PASSIVE  
IDA"U"IDC  
SAD I TTAUP  
CLF 6 /PF 6 TEM STORAGE FOR ALARM CONDITION  
TTLALD, LCH I .  
SAD (770000  
JMP TTAT77 /TWO-BYTE CHAR  
DCH (JMP .+1  
TTFULX, LIO TTTBL+0 /GET OUTPUT CODE  
TCB  
SZF I 6  
JMP TTALM /UNHANG IF HUNG  
LIF  
LAI  
DAP I TTF  
JMP TTLLOOP

/ROUTINE FOR INPUT

TTIN, DAP TTI DCH  
SZF 2  
JMP TT8BIT  
DAP TTI SP  
IDA"U"SZL"U"CLL  
JMP TTI IGN /IGNORE. GENERATED BY SERVICE ROUTINE.  
DAP TTI UP  
TT8INT, LAW TTTBL /ODD INTEGRAL MULTIPLE OF 200  
IAI  
SZF 3  
JMP TTINT /BEING INTERRUPTED  
SAD (TTTBL  
JMP TT1NUL /FIRST NULL  
SZF 5  
JMP TTFULL /NO ROOM IN BUFFER  
DAP . 1  
LIO . /GET WORD FROM TRANSLATION TABLE  
LAC .  
IDC  
TTI UP,  
SAD .  
JMP TTFUL1  
SPI  
JMP TTI 77  
IDA"U"IDC  
SAD I TTI UP  
CLF 6 /PF 6 TEM STORAGE FOR ALARM CONDITION  
TFU1X, LAI  
SAD TTTBL+15  
JMP TTINCR /CR INPUT  
TTINCX,  
TTIDCH,  
RAL 1 S  
DCH I . /PUT CHAR IN BUFFER  
SPA"U"SZF 4  
JMP TTALM /CONTROL MODE ALARM  
RAL 1 S  
SMA"U"SZF 6 I  
JMP TTALM /REGULAR ALARM OR BUFFER NEARLY FULL  
TTI IGN,  
LIF  
LAI  
DAP I TTF  
JMP TTLLOOP

## / BRANCHES AND SPECIAL CASES

/SS 4 UP  
TTSS4, LAW I 1  
SCF"U"AAI  
SPA  
JMP TTSS4X /LOW NUMBER  
LAW 7TT4 /SET SWITCH IN CH 7 ROUTINE  
DAP 7TT2  
RCC /FLUSH. NO TIME TO CKS, SO DEBREAK.  
TTDBRK, LIO TTFLGS  
LFI  
LAC 30  
LIO 32  
JMP I 31

/ROUTINE JSP'ED TO ON CH 7 BREAK  
/SEND RUBOUT ON ALL TT'S TYPEACTIVE WITH LINE CLOSED  
TT4, DAP 7TT2  
LAW 3 /LOWEST TT IGNORED WHEN SS4 IS UP  
DAC 7TTEM  
LSM  
LIA  
SSB  
ADD (TTP"T"200000  
SAL 2S  
DAP • 2  
LAW 41  
AND •  
CLI"U"CMI /RUBOUT IS 377  
SAD (41 /TYPEACTIVE AND LINE CLOSED  
TCB  
IDX 7TTEM  
SAS (100  
JMP 7TT5  
ESM  
JMP 7TT2+1

7 TTEM, 0

/LINE OPEN  
TTLO, LAW TTLBL  
SZL"U"CLL"U"CML"U"IAI  
JMP TTLGC /FINISH CLOSING LINE  
DAP • 2  
LAW I 3777  
AND • /TRANSLATE CHAR  
SAS (360000 /EOM?  
JMP TTLLOOP /NO. IGNORE.  
LIF"U"IAI /ADR PART OF CLEAR  
DAP I TTF  
TTINT1, LIO (215) /START CLOSING LINE OR RUBOUT ECHOED  
TCB  
JMP TTLLOOP

/FINISH CLOSING LINE  
TTLGC, LAW 212 /LF  
CLI"U"SWP"U"STF 6  
TCB  
LIF"U"SCF"U"IAI /LEAVE RING MODE  
DAP I TTF  
LAC (140000+TTLCTX  
JDA TTSOT

/BEING INTERRUPTED BY NULL. PF 3 UP.  
TTINT, SAD (TTTBL"U"377  
JMP TTINT1 /RUBOUT ECHO. SEND CR.  
SAD (TTTBL"U"215  
JMP TTBRK /CR ECHO. SEND LF AND GIVE "BREAK".  
LAC (I  
ADD I TTF  
DIP I TTF /INCREMENT NULL COUNT  
SAS I TTF /END-AROUND CARRY?  
JMP TTLGO /YES. SET LINE TO "OPEN".  
CLI"U"CMI  
TCB  
JMP TTLOOP

/LINE GOING OPEN  
TTLGO, LAW 11  
LIF"U"SCF"U"XAI /CLF 3 AND CLF 6 AND LRG  
DAP I TTF  
LAC (140000+TTLCTX  
JDA TTSOT

/FIRST NULL  
TTINUL, CLI"U"CMI"U"STF 3  
TCB /RUBOUT  
LAC (I /NULL COUNT SET TO "1"  
LIF"U"IAI  
TTSUPX, DAC I TTF /SIC.  
JMP TTLOOP

/ GIVE "BREAK" ALARM AND SEND LF  
TTBRK, DZM HOTFLG  
ISB 1700  
LIO (212 /LF  
TCB  
LAW 3  
ADD TTF  
DAP TTBRK1  
LAW I 1000 /ALL BUT "77 SAVED" BIT  
AND .  
SZF I 1 /FLUSH BUFFER IF OUTPUT  
JMP TTBRKA /INPUT  
DAC I TTBRK1 /OUTPUT  
LIO I TTAUP  
DIO I TTASP  
T TBRKA, AND (377) /KEEP STAT POINTER  
SZA I  
JMP TTSUP /STARTUP PROCEDURE  
SAD BDSTAT  
LAW FSTAT  
DAP TTBRK2  
EEM  
SAD TSTAT /IS IT TOT  
DZM I TOTFLG  
LEM  
LAC (300000  
LSM  
T TBRK2, AND .  
CLI "U"SWP  
SNI  
LAC I TTBRK2  
IOR (40000 /SET MY BIT  
DIP I TTBRK2  
ESM  
LAW 210  
LIF "U"XAI /SET LINK AND CLF 3  
DAP I TTF /LEAVE NON-ZERO NULL COUNT AS "BREAK"  
JMP TTLLOOP

/ STARTUP PROCEDURE  
T TSUP, LAW 7774  
AND TTBRK1  
SAR 2S /GET TT NUMBER+200  
LIA  
LAW WHERE  
DAP TTSUP2  
TTSUP1, IDX TTSUP2 /SEARCH WHERE TABLE FOR 300  
LAW 300  
TTSUP2,  
SAS .  
JMP TTSUP1  
DIO I TTSUP2 /PUT TT NUMBER+200 IN "WHERE" TABLE  
LAW I WHERE-STAT  
ADD TTSUP2  
DAP I TTBRK1 /SET UP PGM WORD  
DAP TTSUP3  
SUB (SAS STAT-WHERE)  
SAD WTOP  
IDX WTOP  
LAW 1 /GIVE HI QUEUE  
TTSUP3,  
DAC .  
LAW 210  
LIF"U"XAI /SET LINK AND CLF 3 AND CLEAR NULL COUNT  
JMP TTSUPX

/ BUFFER ALARM. UNHANG IF HUNG.  
TTALM, CLI "U"STF 6 /RESTORE PF 6  
LAW 3  
ADD TTF  
DAP . 2  
LAW 377  
AND .  
SZA I  
JMP TTALMX /HAS NO PGM. FLUSH ALARM.  
EEM  
SAD TSTAT  
DZM I TOTFLG  
LEM  
SAD BDSTAT  
LAW FSTAT  
DAP TTALM1  
SAS (STAT) /IS IT EXEC DDT  
LIO (1)  
LAW I 7777  
AND .  
SAS (TYOHNG)  
SAD (TYIHNG)  
JMP TTALM2  
SAS (GLPHNG)  
JMP TTALMX  
T TALM1, DIO I TTALM1  
DZM HOTFLG  
T TALMX, LIF "U"IAI /ADR PART OF AC CLEAR  
DAP I TTF  
JMP TTLLOOP

/ TYPEACTIVE CR ECHO. SEND LF. DON'T STORE FLAGS.  
TTACR, LIO (212 /LF  
TCB  
JMP TTLLOOP

/ TYPEACTIVE, GOING PASSIVE: BUFFER EMPTY  
TTEMPTY, LAW 40  
LIF "U"XAI /CLF 1  
DAP I TTF  
JMP TTLLOOP

/ TYPEACTIVE, 12-BIT CHARACTER PROCESSING  
TTA77, LCH I TTALD  
SAS (150000  
SZA I  
CML /IGNORE 7715 AND 7700 ECHOS  
DCH (JMP .+1  
LIO TTBBL+100 /GET TT CODE  
TCB  
LAC I TTASP  
IDA "U"LIF  
SZF 6  
SAD I TTAUP  
JMP TTALM /GIVE ALARM  
LAI  
DAP I TTF  
JMP TTLLOOP

/ 8-BIT MODE INPUT  
TT8BIT, DAP TT8SP  
IDA"U"SZL"U"CLL  
JMP TTIIGN /IGNORE. GENERATED BY SERVICE ROUTINE.  
DAP TTIUP  
IDA  
DAP TT8PGM  
SZF 3  
JMP TT8INT /BEING UNINTERRUPTED  
SNI  
JMP TT1NUL /FIRST NULL. ONLY FULL NULLS WORK.  
RIR 8S  
LAW I 1777 /ALARM CHAR IN TOP 8 BITS OF PGM WORD  
AND .  
TT8PGM,  
XAI  
SZA I /TEST FOR ALARM CHAR  
CLF 6 /PF 6 IS TEM STORAGE FOR ALARM CONDITION  
T8SP,  
LAC .  
IDC  
SAS I TTIUP  
SZF 5  
JMP TTFFULL /BUFFER TOO FULL  
JMP TT8X

/ FULL BUFFER ON INPUT  
TTFULL, CML /IGNORE ECHO  
LIO (334 /BACKSLASH  
JMP TTFULX /SEND CHAR, STORE FLAGS.

/ ALMOST FULL BUFFER ON INPUT  
TTFUL1, SPI /TWO-BYTE CHAR?  
JMP TTFFULL /YES. TOO FULL.  
STF 5 /NO. ROOM FOR THIS AND NO OTHERS.  
JMP TTFLUX

/ 12-BIT CHARACTER PROCESSING ON INPUT  
TTI77, SIR 5S  
TT8X,  
IDC  
SAD I TTIUP  
STF 5 /BUFFER FILLING, WITH PTRS TO BE EQUAL  
IDA  
SAD I TTIUP  
JMP TTI77A /BUFFER FILLING. GIVE ALARM.  
IDC  
SAD I TTIUP  
TTI77A, CLF 6 /PF 6 IS TEM STORAGE FOR ALARM  
LAI  
DCH I TTIDCH /STORE "77"  
JMP TTIDCH

/ CR ON INPUT. SEND LF, TO BE IGNORED ON ECHO.  
TTINCR, CML  
LIO (212  
TCB  
JMP TTINCX

/TYPE PGM, TT, AND "OPEN" OR "CLOSED" ON SOROBAN  
TTSOT, 0 /JDA WITH TEXT PTR  
LAW 3  
ADD TTF  
DAP . 2  
LAW 377 /CALCULATE PGM NUMBER  
AND .  
SUB (-STAT  
SAD (-STAT  
LAW 177  
RCR 9S  
LAW I 3  
EEM  
LSM  
JDA 14SOCT /3 DIGITS AND SPACE  
LAC TTF  
RCR 8S /TTP IS 1000  
LAW I 2  
JDA 14SOCT /2 DIGITS AND SPACE  
LAC TTSOT  
JDA 14STXT  
ESM  
LEM  
JMP TTLOOP

TTLCTX, 464765 //OPEN" WITH CR AND "EOM"  
457756

TTLCTX, 634346 //CLOSED" WITH CR AND "EOM"  
226564  
775600

/CALL SOROBAN OCTAL PRINT IN CORE 15  
14SOCT, 0  
DAP 14SOCT  
LAC 14SOCT  
DAC I 15SOCT  
JSP I 15SOC1  
14SOCT, JMP .

/CALL SOROBAN TOS IN CORE 15  
14STXT, 0  
DAP 14STXX  
LAC 14STXT  
DAC I 15STXT  
JSP I 15STX1  
14STXX, JMP .

TTFLGS, 0

START

DISPATCHER 10/13/66 =DISPAT, 64=

/CHAN 16 BREAK  
DISPAT, LIF"U"SCF"U"CLL  
DIO DSP1 /SAVE FLAGS  
RCL 2S /TOP OF IO CLEAR  
EEM  
SAD ([DDTS6+3]"T"4) /IS IDDT "EXECUTING" INSTR?  
JMP DSP3 /YES, GO ADJUST PC FOR CALLING SEQUENCES  
DSP4, RTB  
DIO TRPBUF  
CLA"U"SWP  
SMA /AN IOT?  
JMP ELGIOT /NO  
RAL 5S /YES, TEST I BIT  
SPA  
JMP DSP16 /YES, CORE 16 IOT  
RAL 1S  
DCH (JMP 140000 DSP2) /REGULAR DISPATCH  
RCL 6S  
LFI  
DSP2, JMP I DSPTB  
DSP3, LAW I 1 /DDT PC ADJUST  
ADD I (DDTS6+4)  
DAP 71  
JMP DSP4  
  
DSP16, RAL 1S /CORE 16 DISPATCH  
DCH (JMP 140000 DSP5)  
RCL 6S /GET REDISPATCH BITS INTO IO  
LFI  
LAC 70 /MOVE AC,PC,IO  
DAC I (C16AC)  
LAC 71  
DAC I (C16PC)  
LAC 72  
DAC I (C16IO)  
LAC DSP1 /FLAGS, TOO  
DAC I (C16FLA)  
LAC I (BITS)  
AND (-40000) /RESTART MODE DEBREAK LEGAL BIT  
DAC I (BITS)  
DSP5, LAC C16TB /GET DEBREAK DISPATCH  
DSP12, DAC 71 /DEBREAK, DISPATCH  
JMP I 71

ILGIOT, EEM  
ELGIOT, LIO I (BITS) /ILLEGAL INSTR.  
LAI  
AND (-500000) /IDDT AND RESTART BITS  
DAC I (BITS)  
SPI I /IS GUY UNDER IDDT  
JMP DSP8 /NO  
LAC C16BKP /BREAKPOINT, GET DDT  
DSP9, LIO 71 /SET UP CRASH AC,PC,IO,C16 TEMP STORAGE  
DAC 71  
DIO I (HH16PC)  
JSP C16TSV  
LIO 70  
DIO I (HH16AC)  
LIO 72  
DIO I (HH16IO)  
LIO DSP1  
DIO I (HH16FG)  
LIO TRPBUF  
JMP I 71

DSP8, RIL 1S  
SPI  
JMP XDDTBK /HELD BY XDDT + CRASHED  
LAC TRPBUF  
SAS (100400) /HLT?  
JMP DSP10  
RIL 3S  
SPI  
JMP REALHALT  
LAC (20000) /REALHALT LEGAL BIT  
IOR I (BITS)  
DAC I (BITS)  
LAC C16HLT  
JMP DSP12

DSP10, LAC C16HH /HALT HORRIBLE  
JMP DSP9

DSP1, 0  
TRPBUF, 0

/HELD BY XDDT + CRASHED; TYPE ON SOROBAN  
XDDTBK, LAW I STAT  
ADD RSTAT  
RCR 9S  
LAW I 3  
LSM  
JDA 14SOCT  
LAW 77  
AND I (TTNO)  
RCR 6S  
LAW I 2  
JDA 14SOCT  
LAC (DCH XDDTB)  
JDA 14STXT  
LAC (CORHNG)  
LEM  
DIP I RSTAT  
JMP HANG1E

XDDTB, 276464  
237756 /XDDT[CR][EOM]

HANG, RCR 6S /HANG GUY WITH STATUS IN AC (12-17)  
JSP HANGS /SUBROUTINE IS IN TT ROUTINES  
HANG1, ISB 1700  
DZM HOTFLG /TELL SWAPPER  
JMP R0

RCK, RCK /REAL RCK  
JMP DPEEK1

DELAY, SZF 1  
JMP CLOCK  
ISB 1700  
DZM I (PRIORITY)  
DELAY1, LAW I 1  
ADD RUNTM  
AND RUNTM /TAKE LOW ORDER BIT FROM RUNTM  
AND (17777)  
SZA I  
JMP R1  
DAC RUNTM  
JMP DELAY1

CLOCK, IDX 71 /HANG FOR NUMBER OF SECONDS IN AC  
LAW 7777  
SUB 70  
DAP RUNTM  
LAW CLKHNG" T "100  
JMP HANG

REALHALT, LEM  
LAC (UNUSED)  
DAC I RSTAT  
LAW WHERE-STAT  
ADD RSTAT  
DAP REALH1  
LAW 300  
REALH1,  
DAC •  
DZM I RCORE  
REALH3,  
LAW I 1  
LSM  
ADD WTOP  
DAP REALH2  
LIA  
LAW 300  
REALH2,  
SAS •  
JMP HANG1E  
DIO WTOP  
ESM  
JMP REALH3  
  
HANG1E,  
ESM  
JMP HANG1  
  
/DEBBREAK FROM CORE 16 THRU HH16  
C16RH,  
LAW 77  
AND I (BILLTT)  
JDA TTNUM  
LAW C16RH1  
DAP TTCKSX  
JMP TTCKS1+2  
C16RH1,  
SAS (2)  
JMP C16RH2  
LAC C16BKK  
DAC 71  
JMP RO  
C16RH2,  
LAC (400000)  
IOR I (BITS)  
DAC I (BITS)  
LIO I (HH16AC)  
DIO 70  
LIO I (HH16PC)  
DIO 71  
LIO I (HH16I0)  
DIO 72  
LIO I (HH16FG)  
JMP RO+1

/RESTART MODE CONTROL IOT  
RSMC, SZF I 5  
JMP RSMC1 /DON'T DEBREAK  
LAC (40000) /DEBREAK LEGAL BIT  
AND I (BITS)  
SZA I  
JMP ELGIOT  
LAW I 1  
ADD I (RESTPC)  
DAC 71  
LIA  
JSP C16TUS /UNSAVE CORE 16 IF NECESSARY

RSMC1, LAC I (BITS)  
AND (-140000) /RESTART MODE AND DEBREAK LEGAL BITS  
SZF 6  
IOR (100000) /RESTART MODE BIT  
DAC I (BITS)  
JMP R1

/ GET A CHARACTER FROM THE READER BUFFER  
/ R1=NOT YOURS, R2=EMPTY, R3=OK, CHAR IN IO

R DA,      DAC RX  
          LAC RSTAT  
          SAS RDWHO  
          JMP I RX  
          IDX RX  
          LAC RB1  
          SAD RB2  
          JMP R4      /READER RUNNING  
          LAC RBI  
          SUB RBO  
          RAL 1S  
          SPA  
          ADD (RDLEN RDLEN)  
          SUB (20)  
          SPA  
R 4,      JMP R5      / START READER ABOUT 20 CHARACTERS IN BUFF  
          LAC RBO  
          SAD RBI  
          JMP I RX      /EMPTY  
          LIO I RBO  
          ADD (400000)  
          SAD RBE  
          SUB (RDLEN)  
          DAC RBO  
          SPA  
          RIR 9S  
          RCR 8S  
          CLI  
          RCL 8S      / IO BITS 10-17  
          IDX RX  
R 5,      JMP I RX  
          LAC RB2  
          DAC RB1  
          RPA  
          RCK  
          DIO RBK  
          JMP R4

R DRLS,      DAC RX      /RELEASE THE READER  
          CLI "U" CMI  
          LAC RSTAT  
          SAD RDWHO  
          DIO RDWHO  
          JMP I RX

/ GET READER ROUTINE  
XRDIN, LIF"U"SCF  
RDINIT, DAC RX  
LAC RDWHO  
SAD RSTAT  
JMP .+3  
SAS (-0)  
JMP RDIN1 /R1, OWNED BY ANOTHER USER  
LAC RSTAT  
DAC RDWHO  
SZF 1  
JMP .+4  
LAC (COR RDRBUF) /GET POINTER TO CORE 15  
DAC RBI  
DAC RBO  
LAC CRB1  
DAC RB1  
IDX RX  
RDIN1, LFI  
JMP I RX  
WRITE P POINTER  
DWPP, LAW I 7 /CHECK TO SEE IF VALID RANGE  
ADD 70  
SZS 50  
JMP DWPP1  
SPA  
JMP ELGIOT  
SUB (40-7)  
SMA  
JMP ELGIOT  
DWPP1, LAC 72  
DAC I 70  
JMP R1  
  
/READ P-POINTER  
DPEEK, LIO I 70  
SZF 1  
DPEEK1, LIO I 72  
SZF 2  
DIO 72  
SZF I 2  
DIO 70  
JMP R1

/ DISPATCH CLEAR THE READER  
DCLR, JSP RDINIT  
JMP R1 / OWNED BY ANOTHER USER  
JMP R2 / OK

/ DISPATCHER RPA  
DRPA, JSP RDA  
JMP ELGIOT / UNOWNED BY USER  
JMP . 3  
DIO 72  
JMP R1 / RETURN TO USER WITH CHAR IN IO  
LAW RPAHNG"T"100  
JMP HANG

/ DISPATCHER RELEASE THE READER  
DRLSRD, JSP RDRLS  
JMP R1

/ DISPATCHER GET THE PUNCH  
DGETPU, LAC PUNWHO  
SAD RSTAT  
JMP R2  
SAS (-0)  
JMP R1 / OWNED BY ANOTHER USER  
LAC RSTAT  
DAC PUNWHO  
JMP R2

/ DISPATCHER RELEASE THE PUNCH  
DRLSPU, CLI "U" CMI  
LAC RSTAT  
SAD PUNWHO  
DIO PUNWHO  
JMP R1

/ DISPATCHER PPA  
DPPA, LAC RSTAT  
SAS PUNWHO  
JMP ELGIOT  
LAC PUP  
IDA  
SAD PEND  
SUB (PUNLEN)  
SAD PSP  
JMP PFULL  
LIO 72  
DIO I PUP  
DAC PUP  
LSM  
LAC PRUN  
SZA  
ISB 1200  
ESM  
JMP R1  
P FULL, LAW PPAHNG"T"100  
JMP HANG

/ IOT TO START A USER  
P SUC, ESM  
P SU, LEM  
LAW WHERE-1  
DAP PSUA  
IDX PSUA  
LAW 300  
P SUA, SAS .  
JMP N-3  
LIO (500)  
LSM  
SAS I PSUA  
JMP PSUC  
DIO I PSUA  
LAW 7777  
AND PSUA  
SAD WTOP  
IDX WTOP  
ESM  
LAW I WHERE-STAT  
ADD PSUA  
DAP PSUB  
LAW 1  
DAC .  
DZM HOTFLG  
JMP R1

## / DISPATCHER TT ROUTINES

/ SET UP PTRS USED BY DISPATCHER TT ROUTINES

TTSET, DAP TTSETX  
 LAW 77  
 AND I TTNO  
 JDA TTNUM  
 JMP .

TTNUM, 0 /JDA WITH TT NUMBER IN AC  
 DAP TTNUMX  
 LAW TTP"T"200000  
 ADD TTNUM  
 SAL 2S /GET PLACE IN TTP TABLE  
 DAP TTFP  
 IDA  
 DAP TTSP  
 IDA  
 DAP TTUP  
 DAP TTDCHU  
 DAP TTLCHU  
 IDA  
 DAP TPPP  
 JMP .

TTFP, 140000+.  
 TTSP, 140000+.  
 TTUP, 140000+.  
 TPPP, 140000+.

/FLAGS  
 /SERVICE PTR  
 /USER PTR  
 /PROGRAM

/ CHECK TT STATUS IOT  
 TTCKS, JSP TTCKS1

SZA I  
 JMP TTRETN  
 DAC 70 /PUT ERROR CODE IN AC AND ERROR CODE WORD  
 JMP TTCKS2 /CLEAR OUT BREAK IF ANY

SUBROUTINE TO CHECK OWNERSHIP AND BREAK

TTCKS1, DAP TTCKSX  
 JSP TTSET /TT NUMBER IS IN USER CORE  
 TTCKS1+2, LAW 377  
 AND I TPPP  
 LIO 1  
 SAS RSTAT  
 JMP TTCKSY /NOT HIS. ERROR CODE 1.  
 LAW I 7767 /770010 TO AC  
 AND I TTFP  
 RAR 4S  
 RIL 1S  
 SPQ /BREAK?  
 CLI /NO  
 TTCKSY, LAI  
 TTCKSX, JMP .

/ SUBROUTINE NORMALLY USED BY TT IOT'S. CHECKS OWNERSHIP  
 / AND DETECTS "BREAK" STATUS AND CLEARS IT. GIVES TRAP (OR R1)  
 / UNLESS EVERYTHING OK.

T TOK, DAP TTOKX  
 JSP TTCKS1  
 SZA I  
 T TOKX, JMP .  
 TTCKS2, SAS '2  
 JMP TTERR /NOT HIS TT  
 DHANGX, LAW 10 /BREAK. CLEAR OUT BREAK STATUS  
 LSM  
 AND I TTFP  
 SZA I  
 DIP I TTFP /CLEAR NULL COUNT UNLESS BEING INTERRUPTED  
 ESM  
 LAW 2 /ERROR CODE FOR BREAK  
 DAC I (ERCODE) /SAVE ERROR CODE  
 LIO 71 /SAVE ADDRESS OF ERROR IN TRAPPC  
 DIO I (TRAPPC)  
 SZF 6  
 JMP R1  
 LAW TTTSU  
 SPI  
 IOR (400000) /SET OVERFLOW IF PREVIOUSLY ON  
 DAC 71  
 JMP R0

/ IOT TYI. TYPE IN CHAR TO TOP 6 BITS OF AC.  
 TYI=. JSP TTOK  
 JSP TTU /GET CHAR OR HANG  
 DAC 70 /STORE CHAR  
 JMP TTRETN

/ IOT TYO. TYPE CHAR IN TOP 6 BITS OF AC.  
 TYO=. JSP TTOK  
 LAW I 7777  
 AND 70  
 TTRETN, JDA TTS /TYPE CHAR OR HANG  
 SZF 6 /TEST WHETHER 1- OR 2-RETURN IOT

R2, IDX 71 /NORMAL RETURN 2  
 R1, IDX 71 /NORMAL RETURN 1  
 R0, LIO DSP1 /NORMAL RETURN 0  
 R0+1, LFI /RESTORE FLAGS  
 LAC 70  
 LIO 72  
 JMP I 71

/IOT TIS. TYPE IN STRING TERMINATED BY 74. AC IS PTR.  
 TIS,  
 JSP TTOK  
 LAC I (TISMAX  
 IDC  
 DAC DTEM1  
 TIS1,  
 LAC 70 /LOOP. TEST FOR VALID PTR.  
 IDC"U"SCI  
 SAD DTEM1  
 JMP TIS2 /TISMAX EQUALLED. WILL BE EXCEEDED.  
 AND C177740  
 RCL 6S  
 SNI"U"SZA I  
 JMP TIS3 /TOO LOW OR TOO HIGH  
 JSP TTU /GET CHAR OR HANG  
 DCH I 70  
 SAD C74  
 JMP TTRETN /TERMINATOR SEEN  
 JMP TIS1

TIS2,  
 LAW 3 /TISMAX ERROR CODE  
 JMP TTERR

TIS3,  
 LAW 4 /ILLEGAL REGION ERROR CODE

/IOT TOS. TYPE STRING TERMINATED BY 74. AC IS PTR.  
 TOS,  
 JSP TTOK  
 TOS1,  
 LIO 70  
 LCH I 70 /GET CHAR  
 SAD C740000  
 JMP TTRETN /RETURN, LEAVING AC STEPPED.  
 DIO 70 /UNSTEP IN CASE OF HANG  
 JDA TTS /TYPE CHAR OR HANG  
 LCH I 70 /STEP AC  
 JMP TOS1

/SUBROUTINE TO TAKE 6-BIT CHAR OUT OF BUFFER. HANG USER IF EMPTY.  
 TU,  
 DAP TTSX /EXIT THROUGH TTS  
 ERG  
 LAW I 2  
 DSC 600  
 AND I TTFP  
 SAS I TTFP  
 JMP TTUFUL /PF 5 SET. BUFFER FULL WITH PTRS EQUAL.  
 RCR 6S /SAVE PF 1 IN I.O.  
 LAC I TTUP  
 SPI I  
 SAD I TTSP  
 JMP TIHANG /TYPEACTIVE?  
 TTU1,  
 IDC /EMPTY?  
 DAC I TTUP /TYPEACTIVE OR BUFFER EMPTY  
 JMP TTSS /STEP USER PTR

TTUFUL,  
 DAC I TTFP  
 LAC I TTUP /CLEVERLY CLEARS PF 5  
 JMP TTU1

/ SUBROUTINE TO TYPE 6 BITS IN TOP OF AC. HANG USER IF BUFFER FULL.

T TS, 0  
 DAP TTSX  
 LAW I 7777  
 SAD TTS  
 JMP TTS7 /WARNING CHAR. GOBBLE IT UP.  
 ERG  
 LAW 40  
 LIO I TTUP  
 DSC 600  
 IOR I TTFP /STF 1  
 SAS I TTFP /ALREADY SET?  
 JMP TTSBEG /NOT TYPEACTIVE. START HIM UP.

T TS1,  
 LAI  
 IDC  
 SAD I TTSP  
 JMP TOHANG /BUFFER FULL  
 LIO I TPP  
 RIL 8S  
 SPI I  
 JMP TTS2 /77 NOT SAVED UP  
 IDC  
 SAD I TTSP  
 JMP TOHANG /NO ROOM FOR 2-BYTE CHAR  
 CLC

T TS2,  
 TTDCHU,  
 T TS3,  
 LAI 1000  
 AND I TPP  
 DAC I TPP /SET OR CLEAR "SAVED 77" STATUS  
 SZL I /SKIP IF EXEC DDT  
 IDX I NOCHAR /STEP COUNT OF CHARS HANDLED  
 JSP ASC6 /RE-ACTIVATE CH 6  
 T TTLCHU,  
 LCH . /GET CHAR FOR TTU RETURN  
 LRG

T TSX,  
 JMP .

T TSBEG,  
 DIO I TTSP /SET PTRS EQUAL  
 AND C-2 /CLF 5  
 DAC I TTFP  
 AND C211 /KEEP LINK, PF 3, AND PF 6  
 SAS C1  
 JMP TTS1 /EXPECTING A BREAK: DON'T TCB  
 LIO TTNUM  
 SSB  
 LAW TTBLLT"10000" /CALCULATE PLACE IN TBL  
 IOR TTS  
 RAL 6S  
 LIO I TPP  
 RIL 8S  
 SPI /"SAVED 77" KEPT IN "1000" BIT  
 ADD C100 /77 SAVED UP

DAP • 1  
LIO : /GET CODE  
TCB : /SEND IT  
SAS (TTTBL+115  
SAD (TTTBL+100  
JMP TTS6 /BREAK OR CR-NO-LF. SET LINK.  
JMP TTS3

TTS6, LAW 200 /SET LINK  
IOR I TTFP  
DAC I TTFP  
JMP TTS3

TTS7, LAW 1000 /SET "SAVED 77" STATUS  
IOR I TPPP  
JMP TTS4

/ROUTINES TO HANG USER, ETC.  
TIHANG, LIO (TYIHNG  
JMP TTHANG

TOHANG,  
TTHANG, LIO (TYOHN  
LRG  
IDX TTSX /IN CASE XDDT  
SZL  
JMP TTSX /XDDT. GIVE HIM R2.  
SZF 5  
JMP TTHNG1 /A "DON'T HANG ME" IOT  
JSP HANGS /HANG USER  
JSP ASC6 /REACTIVATE CHANNEL  
JMP HANG1

G TYBSX,

/ SUBROUTINE TO HANG USER. HUNG STATUS IN I.O.  
HANGS, DAP HANGSX  
LEM  
LAW I 7777 /LOOK AT TOP 6 BITS OF...  
LSM  
AND I RSTAT /...GUY'S STATUS  
SZA I  
DIO I RSTAT /NO SPECIAL BITS SET; HANG HIM  
ESM /N.B. GUY'S QUEUE IS IN RUNTM  
HANGSX, JMP .

/ SUBROUTINE TO RE-ACTIVATE CH 6 AND ISB IF SCANNER STOPPED.  
ASC6, DAP ASC6X  
ASC 600  
CKS  
RIR 2S  
SPI  
ISB 600 /SCANNER STOPPED  
ASC6X, JMP .

/ IOT'S TO LEAVE, ENTER, AND SAVE UP 8-BIT MODE AND CONTROL MODE.

T TMODE, JSP TTCKS1  
SAD 1  
JMP TTERR /NOT HIS  
LAC I TTFP  
SZF 3  
DAC 72 /RETURN OLD STATUS IN I.O.  
LAW 4  
SZF I 1 /8-BIT IOT?  
JMP TTMOD1 /NO  
LAC 70 /YES. SET NEW ALARM CHAR.  
XOR I TPP  
AND C776000 /ALARM CHAR IN TOP 8 BITS  
XOR I TPP  
LIO I TPP  
DAC I TPP  
SZF 3  
DIO 70 /PUT OLD ALARM CHAR IN AC  
LAW 20  
DAC DTEM1 /STORE FLAG BIT FOR MODE  
CMA  
LSM  
AND I TTFP  
SZF 2 /ENTER MODE?  
IOR DTEM1 /YES  
DAC I TTFP  
ESM  
JMP TTRETN

/ IOT "GET TELETYPE"

G TY, JSP TTSET  
LIO RSTAT  
LAW 377  
LSM  
AND I TPP  
SZA I  
DIO I TPP /IF NO ONE'S: ASSIGN TO HIM  
ESM  
SZA  
SAD RSTAT  
JMP TTRETN /NORMAL RETURN  
T THNG1, CLC /"DON'T HANG ME" CODING  
DAC 72 /-0 IN I.O. TO INDICATE HANG CONDITION  
JSP ASC6 /REACTIVATE CHANNEL 6  
JMP TTRETN /GIVE HIM A GOOD RETURN

/IOT RELEASE TT

RTY, JSP TTCKS1  
SAD (1) /NOT YOUR TT?  
JMP TTER  
DZM HOTFLG  
DZM I TOTFLG  
LAW 7741  
LSM  
AND I TTFP  
DAC I TTFP  
DZM I TPP  
ESM  
LIO (1)  
LAG TSTAT  
LSM  
SAD BDSTAT  
LAW FSTAT  
DAP RTY1  
LAW I 7777  
AND I RTY1  
SAD (GLPHNG)  
DIO I RTY1  
JMP TTMODX

RTY1, DCH .

/ IOT TO HANG USER UNTIL BREAK ON TT IN TTNO  
DHANG, DSC 600  
JSP TTCKS1  
LIO (EOOTHNG)  
SAS (2) /BREAK?  
JMP TTHANG /NO. HANG HIM  
JSP ASC6  
JMP DHANGX /IN TTOK, CLEAR OUT BREAK AND UNHANG

/ EXEC DDT TYO AND TYI  
X DDTYO, JDA XDDTX /CALLED FROM CORE 10  
CLA  
RCR 6S  
JDA TTS /TYPE FROM BOTTOM OF IO  
JMP XDDTXR  
X DDTXY, LAI  
LIO DDTFLG  
LFI  
LIA  
JSP HNGDDT  
X DDTX1, CLL"U"CML /XDDT FLAG FOR TT ROUTINES  
LAC EXECTT  
JDA TTNUM  
LIO XDDTEM  
JMP .  
X DDTXX, 0 /SETUP POINTERS  
DAP XDDTXX  
DIO XDDTEM  
LIF"U"SCF  
DIO DDTFLG  
JMP XDDTX1

X DDTXR, LIO DDTFLG /RETURN ACROSS CORE  
LFI  
LIO XDDTEM  
JMP I XDDTX

X DDTYI, JDA XDDTX /CALLED FROM CORE 10  
JSP TTU  
JMP XDDTXR  
DZM DDTGUY  
JMP XDDTXY

X DDTEM,  
EXECTT,  
DTEM1, 0  
0  
0

/ROUTINE FOR DISPATCHER TO HANG XDDT  
HNGDDT, DAC HDDTX  
LAW I 7777  
LSM  
AND STAT 0 /XDDT IS USER 0  
SWP  
SNI  
DIP STAT 0  
ESM  
JSP ASC6 /RE-ACTIVATE CH 6 AFTER HANGING DDT  
LIF  
DIO DDTFLG  
ESWORG,  
LEM  
JMP SWORG /ON CH 17 BREAK - CALLED FROM XDDT

/ROUTINE TO START UP XDDT WHEN IT COMES INTO CORE  
RUNDDT, LAW I 7777  
LSM  
AND STAT 0  
SZA  
JMP DDTBRK /BREAK KEY ON EXEC TT  
ESM  
LAC RCORE  
DAC DCORE  
SAR 2S  
IOR (200000) /XDDT RUNS IN CORE 10 (RENAMED)  
DCH (ADD .+1)  
RNM  
LAC DDTGUY  
SZA /DID HE HAVE ANYBODY  
JMP DDTWT1 /YES, GET HIM INTO CORE AGAIN  
EEM  
LIO DDTFLG  
LFI  
JMP I HDDTX /NO, GO BACK TO DDT'S CORE

HDDTX, 100000  
DDTGUY, 0  
DDTSU, 100000  
DDTNUS, 100001  
DDTFLG, 0

DDTBRK, LAW 1  
DAC STAT 0  
LAC DDTSU  
DAC HDDTX  
DZM DDTGUY  
JMP RUNDD1

/ROUTINE FOR XDDT TO CALL TO GET A GUY INTO CORE  
D DTWNT, DAC HDDTX  
LEM  
LAW STAT  
AAI  
DAC DDTGUY  
DDWT1, LAW WHERE-STAT  
ADD DDTGUY  
DAP DDTWT2  
LAW 777  
LSM  
DDWT2, AND .  
SAD (300 /IS THERE SUCH A GUY  
JMP DDTHLT /NO  
LAC (300000)  
AND I DDTGUY  
CLI "U" SWP  
SNI  
LAC I DDTGUY  
IOR (10000)  
DIP I DDTGUY /SET BIT  
ESM  
LAW I 2  
DAC I DCORE /HOLD DDT'S CORE  
JMP HNGDD2

/ROUTINE TO START UP XDDT WHEN HIS USER COMES INTO CORE  
DDTWS, LAC (-10000) /REMOVE XDDT WANTS BIT  
LSM  
AND I RSTAT  
DIP I RSTAT  
ESM  
LAC RSTAT /IS THIS GUY DDT CURRENTLY WANTS  
SAS DDTGUY  
JMP SWORG /NOT GUY: FORGET IT  
LAW STAT 0 /IS GUY: SET DDT RUNNING  
DAC I DCORE  
DAC RSTAT  
JMP RUNDD2

DDTHLT, ESM  
LAC DDTNUS /TELL DDT THAT THERE'S NO USER  
DAC HDDTX  
JMP RUNDD2

/ XDDT RPA  
D DTRPA, DAC DDTRPX  
D DTRP1, JSP RDA  
XX  
JMP DDTRDH  
JMP I DDTRPX  
D DTRPX,  
O  
D DTRDH, LIO (RPAHNG)  
JSP HNGDDT  
JMP DDTRP1

S TART XX-JMP

SWAPPER 10/13/66 =SWAP, 56=

/PRESWAPPER, HANDLES RUN TIME FOR RUNNING USER

SWAPT, 0 /TIME OF LAST DEBREAK (IN MS.)

SWAP, DSC 1700 /BREAK ON CHAN 17

SZS 10

JSP CEXECCTT

EEM

RCK

LAI

SUB SWAPT /CALCULATE ELAPSED TIME

SPA

ADD (60000.)

DAC SWAPT /SAVE ELAPSED TIME

LIF"U"SCF"U"CLL /PRESERVE LINK IN IO

TAD I (TOTRT+1)

DAC I (TOTRT+1)

SZL"U"LFI

IDX I (TOTRT)

LAC (700000)

LEM

AND I RSTAT

EEM

SZA

SAD (400000)

JMP SWAPH

DAP I (BITS)

JMP SWAPI

SWAPH,

LAW 7777

AND I (BITS)

ADD SWAPT

LIA

AND (37)

DAP I (BITS)

XAI

SZA I

JMP SWAPI

LAW I I

ADD RUNTM

LIA

AND RUNTM

AND (1777)

SWP

SNI I

JMP SWAPA

LAW 7400 /GUY CHANGING QUEUE

AND RUNTM /IS HE BACKGROUND?

SZA

JMP SWAPD /YES

LAC RUNTM

SAL 2S

SUB. (1)

S WAPG, DZM HOTFLG  
S WAPA, DAP RUNTM /UPDATED RUN TIME  
S WAPI, LIO HOTFLG  
S WAPC, SNI  
JMP SWAPB  
SWDBK, LAC RSTAT  
SAS BDSTAT  
SZS I 30  
JMP SWDBK1

S WAPB, LAC 74  
DAC I (AC) /SETUP P POINTERS FROM DEBREAK AREA  
LIO 75  
DIO I (PC)  
JSP C16TSV  
LIO 76  
DIO I (IO)  
LIF"U"SCF"U"CLL  
DIO I (FLAGS)  
LEM /SWAPPER RUNS OUT OF EM  
LAC RUNTM  
DAP I RSTAT

## /\* SWAPPER PROPER

SWORG, CLC"U"CLF 7  
 DAC HOTFLG /INIT. HOTFLG  
 LAW I 7777 /SEE IF XDDT PERMANENTLY HUNG  
 AND STAT 0  
 SAS (40000)  
 JMP SWLO  
 LAW I 2  
 SAS I DCORE  
 JMP SWLO  
 LAW STAT 0  
 DAC I DCORE  
 DZM DDTGUY  
 SWLO, CLA  
 SAS LDSTAT  
 STF 1  
 SAS BDSTAT  
 STF 2  
 SWLOA, LAW STAT /INIT TO FIND BEST GUY  
 DAP SWV1  
 LAW WHERE  
 DAP SWV2  
 LAC (I)  
 DAC SWBQ  
 SWV1, LIO . /GET HIS QUEUE  
 SPI /HUNG?  
 JMP SWL1 /YES, IGNORE  
 LAW 777 /NO  
 SWV2, AND . /FIND OUT WHERE HE IS  
 RAR 6S  
 ADD (SWSTB) /LOOK UP IN SKIP TABLE  
 DAP SWV3  
 SWV3, XCT . /SKIP IF ACCESSIBLE  
 JMP SWL1 /INACCESSIBLE, IGNORE  
 CLA /FIGURE EFFECTIVE QUEUE  
 RCL 6S  
 SZA  
 CLI  
 RIR 6S  
 LAI  
 SUB SWBQ  
 SMA  
 JMP SWL1 /NOT AS GOOD AS BEST SO FAR  
 DIO SWBQ /BETTER, UPDATE BEST SO FAR  
 LAC SWV2  
 DAP SWV7  
 SWL1, IDX SWV1 /LOOK AT NEXT GUY  
 IDX SWV2  
 SUB (AND)  
 SAS WTOP  
 JMP SWV1  
 LAC SWBQ  
 SAD (I) /DID WE FIND ANYBODY?  
 JMP SWBGC /NO, RUN BEST GUY IN CORE  
 CLC  
 DAC SWWQ  
 LAW CS1  
 DAP SWV4  
 SWV4, LAC . /GET CORE STATUS  
 SPA  
 JMP SWL2 /UNAVAILABLE, IGNORE  
 SZA I

JSR 5  
 DAP 10  
 JMP 5

JMP SWL3 /FREE, STOP SEARCHING

SWV5, DAP SWV5  
 LIO . /LOOK AT STATUS OF PGM IN THAT CORE  
 SPI  
 JMP SWL9 /HUNG, EFFECTIVE QUEUE IS (1)  
 CLA  
 RCL 6S  
 SZA  
 CLI  
 RIR 6S  
 SWL10, LAI /GET STATUS INTO AC TOO  
 SUB SWWQ SPA ~~SPQ~~? AH! So!  
 JMP SWL2 /ALREADY HAVE A WORSE GUY  
 DIO SWWQ /REMEMBER NEW WORSE GUY  
 LAC SWV4  
 DAP SWT2  
 IDX SWV4 /LOOK AT NEXT CORE  
 SAS (LAC CS3+1)  
 JMP SWV4  
 LAC SWWQ  
 SPA  
 JMP SWNN /DIDN'T FIND ANY CORES AT ALL, WAIT  
 SUB SWBQ  
 SPQ /SWAPPABLE PAIR?  
 JMP SWBGC /NO, GO RUN BEST GUY IN CORE  
 LAW 700 /YES, WHERE IS NON-CORE GUY?  
 SWL8,  
 SWV7, AND .  
 CLI  
 SZA I  
 JMP SWLD /LITTLE DRUM  
 SAD (100)  
 STF 4 /FG 4 ON MEANS B.D., OFF MEANS LIMBO  
 SZF 3 /FG 3 INDICATES FREE CORE  
 JMP SWFC  
 LAW WHAT /INIT. TO SEARCH FOR FREE L.D. FIELD  
 DAP SWV6  
 SWL5, LAW 777  
 SWV6, AND .  
 SAD (400)  
 JMP SWL4 /FOUND FREE L.D. FIELD  
 IDX SWV6  
 SAS (AND WHAT+100)  
 JMP SWL5  
 SZF 4  
 JMP SWL11 /GUY IS ON BIG DRUM  
 DAP SWV11 /GUY IS IN LIMBO, INIT. TO FIND B.D. FIELD  
 LAW 777  
 SWV11, AND .  
 SAD (400)  
 JMP SWL11A /FOUND FREE B.D. FIELD  
 IDX SWV11  
 SAS (AND WHAT+200)  
 JMP SWL12 /KEEP LOOKING

/ SORRY, YOU LOSE.  
LIO I SWV7  
LAW I WHERE-STAT  
ADD SWV7  
DAP SWV11A  
LAW 300  
LSM  
DAC I SWV7  
LAC (UNUSED)  
SWV11A,  
DIP .  
LAW 200  
XAI "U"CLL"U"CML /WE'LL NEED THE LINK LATER  
SAS (700)  
JMP SWV11C  
EEM  
LAC (DCH SWV11Z)  
JDA 14STXT  
LEM  
SSWORG,  
ESM  
JMP SSWORG  
SWV11C,  
JDA TTNUM /SET UP TT POINTERS  
DZM I TPP /FLUSH THE PROGRAM NUMBER  
LAW SWV11Y  
DAC SWV11W  
SWV11D,  
LCH I SWV11W  
SAD (SPO)  
JMP SSWORG  
JDA TTS /GO TYPE OUT THE CHARACTER  
NOP  
JMP SWV11D  
SWV11W,  
0  
SWV11Z,  
472224  
004471  
222265  
647756  
SWV11Y,  
TEXT /CALL LATER PLEA/  
634577  
157712  
770474

/MISCELLANEOUS BITS OF CODING  
/FOUND FREE CORE  
SWL3, STF 3 /FREE CORE INDICATOR  
LAC SWV4  
DAP SWT2 /REMEMBER WHICH CORE  
JMP SWL8

/SKIP TABLE FOR FIGURING ACCESSIBILITY  
SWSTB, SZF 1 /ON L.D.  
SZF 2 /ON B.D.  
SZF 6 /IN LIMBO  
NOP /HALTED PGMS ARE NEVER ACCESSIBLE  
NOP /CORE IS NEVER ACCESSIBLE  
SZF 6 /STARTUP PROGRAM PRIME

/PGM IN CORE IS HUNG  
SWL9, LIO (I)  
JMP SWL10

SWV20, LIO WHAT /VARIABLE FOR SWL11 LOOP

/FIND GUY TO PUT ON B.D.  
SWL11A, SZF 1 2  
JMP SWL11  
STF 6  
JMP SWL0A

SWL11, LAC SWV20  
DAP SWV13  
CLC  
DAC SWWQ

SWV13, LIO . /SEARCH ONLY LD WE HAVE  
LAW 700  
NAI  
SZA  
JMP SWL14 /NOT A GUY THERE  
LAW STAT  
AAI

SWL13, DAP SWV12  
SAD (STAT)

SWV12, JMP SWL14 /DON'T SWAP XDDT ON FASTRAND  
LAC . /GET GUY'S STATUS  
LIA /INTO IO  
AND (720000)  
SAS (400000)  
SAD (020000)  
JMP SWL14 /SKIP ANY FASTRAND GUYS  
SZA  
CLI  
SPA  
LIO (I) /EXCEPT HUNG GUYS  
LAI /EFFECTIVE Q IN AC, IO  
SUB SWWQ  
SPA  
JMP SWL14 /WORSER GUY ALREADY (SIC)  
DIO SWWQ /REMEMBER THIS GUY

LAW WHERE-STAT  
 ADD SWV12  
 DAP SWV16  
 SWL14,  
 SZF 6 /GET NEXT GUY, CORE OR LD?  
 JMP SWL15 /CORE, GO LOOK AT NEXT CORE  
 IDX SWV13 /LOOK AT NEXT LD FIELD  
 SAD (LIO WHAT+40)  
 LAC (LIO WHAT)  
 DAP SWV13  
 SAS SWV20  
 JMP SWV13  
 LAW WHAT  
 ADD I SWV16  
 DAP SWV20  
 STF 6 /NO LD FIELDS LEFT, LOOK AT CORES NOW  
 LAW CS1  
 DAP SWV15  
 LAC . /CORE STATUS WORD  
 SZM /IS CORE OK?  
 JMP SWL13 /YES, COMPARE GUY WITH OTHERS  
 IDX SWV15 /GET NEXT CORE  
 SAS (LAC CS3+1)  
 JMP SWV15  
 LAC SWWQ  
 SPA  
 JMP SWBGC /NO, WAIT  
 SUB SWBQ  
 SPQ /FASTRAND SWAPPABLE PAIR?  
 JMP SWBGC  
 LAW 400 /YES, WHERE IS HE?  
 AND . /WHERE PTR TO GUY FOUND  
 SZA  
 JMP SWBD /CORE, GO SET UP BD WRITE  
 SAS LDSTAT /LD, SETUP LD READ UNLESS LD BUSY  
 JMP SWBGC /LD BUSY, WAIT  
 LAW WHAT  
 ADD I SWV16  
 DAP SWV13  
 SWL17,  
 LAW I WHAT  
 ADD SWV13 /GET L.D. READ FIELD  
 DAP SWT3  
 RAR 6S /INTO HI ORDER AC  
 SPA /WHICH L.D.?  
 STF 5  
 IOR (400000) /SET READ BIT  
 DIP LDIA  
 LAW 777  
 AND I SWV13 /GET INCOMING PGM  
 ADD (STAT)  
 DAP LDT2 /SETUP FOR L.D. S.B. ROUTINE  
 ADD (WHERE-STAT)  
 DAP LDV1  
 LAW I CS1-1-400 /GET PHYSICAL CORE  
 ADD SWT2  
 DAC LDT1  
 RAR 4S  
 DIP LDCL  
 LAC SWT2  
 DAP LDV2

SZF 3  
JMP SWL19 /READ ONLY, GO DO DIFFERENT THINGS.  
LAW 201 /SAVE L.D. READ FIELD FOR NEXT SWAP  
DAC I SWV13  
LIO SWT3 /GET READ FIELD  
SZF 5 /WHICH DRUM?  
JMP SWL20  
LAC SWFLDF /DRUM 0, RESET SAVED L.D. FIELD  
DIO SWFLDF  
JMP SWL21  
  
SWL20,  
LAC SWFLDF+1 /DRUM 1, RESET SAVED FIELD  
DIO SWFLDF+1  
SWL21,  
LIA /GET WRITE FIELD IN AC AND IO  
RAR 6S  
SWL24,  
IOR (400000) /SET WRITE BIT, WRITE ONLY JOINS HERE  
DIP LDWC  
LAW WHERE-STAT  
ADD I SWT2 /GET WHERE PTR. TO GUY BEING WRITTEN  
DAP .+1  
DIO . /SAY HE'S ON THE DRUM  
LAW WHAT /GET WHAT PTR. TO WRITE FIELD  
AAI  
DAP SWV17  
LAW I STAT /SAY DRUM FIELD HAS GUY  
ADD I SWT2  
SWV17,  
DIO .  
SWL23,  
CLC /SET UP STATUS FOR SWAPPING  
DAC LDSTAT  
DAC I SWT2 /SAY CORE IS HELD  
LAW I 2  
DAC LDERC  
SZF 5 /WHICH DRUM?  
JMP SWL22

LSM  
 DRA  
 LAW 48.  
 AAI  
 DAP LDIA  
 DAP LDCL  
 LIO LDIA  
 DIA  
 LIO LDWC  
 DWC  
 LIO LDCL  
 DCL  
 ESM  
 CLA /SETUP SKIP FOR SEQ.BRK. ROUTINE  
 SWL25, DAP LDV3  
 JMP SWORG

SWL22, LSM /SAME ROUTINE FOR DRUM 1  
 DRA 100  
 LAW 48.  
 AAI  
 DAP LDIA  
 DAP LDCL  
 LIO LDIA  
 DIA 100  
 LIO LDWC  
 DWC 100  
 LIO LDCL  
 DCL 100  
 ESM  
 LAW SPA"U"SMA-SKP  
 JMP SWL25

/NORMAL ENTRY TO LITTLE DRUM CODING  
 SWLD, LAW WHAT  
 ADD I SWV7 /GET READ FIELD  
 DAP SWV13  
 JMP SWL17 /ENTER L.D. CODING

/READ ONLY CODING  
 SWL19, LAW 400  
 DAC I SWV13 /MARK READ FIELD AS FREE  
 CLA /SETUP TO NOT WRITE  
 DIP LDWC  
 JMP SWL23 /GO DO DRUMSWAP

/WRITE ONLY CODING  
 SWL18, CLA /ENTRY POINT. SETUP TO NOT READ  
 DIP LDIA  
 LAC SWT2 /GET CORE TO SET UP FOR S.B. ROUTINE  
 DAP LDV2  
 SUB (CS1-1) /GET PHYSICAL CORE  
 RAR 4S  
 DIP LDCL  
 LAC SWV6  
 SUB (AND WHAT) /GET WRITE FIELD  
 LIA  
 RAR 6S  
 SPA /WRITE FIELD DETERMINES DRUM  
 STF 5 /DRUM INDICATOR  
 JMP SWL24 /ENTER MAINLINE SETUP CODING

/LITTLE DRUM SEQUENCE BREAK ROUTINE

L D, DRA 100  
 L DV3, SKP /SKIP IF DRUM 1 IS BEING USED  
 DRA /GET DRA ON CORRECT DRUM  
 LAC . /10 MICROSECS.  
 SPI  
 JMP LDERR /ERROR ROUTINE  
 DZM LDSTAT /RESTORE STATUS, UNAFFECTED BY EM  
 DZM HOTFLG /MARK ALARM  
 ISB 1700 /ACTIVATE SWAPPER  
 LAC LDIA  
 SMA /WRITE ONLY?  
 JMP LDWO /YES  
 L DV1, LIO LDT1 /PRE SET-UP CORE OF GUY WHO CAME IN  
 DIO . /MARK GUY WHO CAME IN AS BEING IN CORE  
 L DT2, LAW . /STAT PTR. FOR GUY WHO CAME IN  
 L DV2, DAC . /RESTORE CORE STATUS  
 L DERGI, LAC 24  
 LIO 26  
 JMP I 25 /DEBREAK  
 L DWO, CLA /SET CORE TO FREE  
 JMP LDV2

L DT1, 0

L DERR, RIL 1S  
 SPI I  
 XX /TIMING ERRORS  
 ISP LDERC  
 JMP LDERG /TRY AGAIN  
 LAW I STAT  
 ADD LDT2  
 RCR 9S  
 LAW I 3  
 EEM  
 LSM  
 JDA 14SOCT  
 LAC (LDERT)  
 JDA 14STXT  
 LAC LDIA  
 XCT LDV3  
 AND (-ADD)  
 LIA  
 LAW I 2  
 JDA 14SOCT  
 LAC (LDECRT)  
 JDA 14STXT  
 ESM

JMP LDERR  
 L DECRT: 775600  
 L DERT: 436400  
 476151  
 005600

L DERG, 0

L DERG, CLA  
DIP LDWC  
LAC LDIA  
SMA  
JMP LDWO  
XCT LDV3  
JMP LDERGO  
LIO LDIA  
DIA 100  
LIO LDWC  
DWC 100  
LIO LDCL  
DCL 100  
JMP LDERGI

L DERGO, LIO LDIA  
DIA  
LIO LDWC  
DWC  
LIO LDCL  
DCL  
JMP LDERGI

/ FOUND FREE L.D. FIELD  
 SWL4, CLA /IS L.D. BUSY?  
 SAD LDSTAT  
 JMP SWL18 /NO, DO DRUM WRITE TO GET FREE CORE

/ THIS SECTION FINDS THE BEST GUY IN CORE TO RUN  
 SWBGC, LAW CS1 /INITIALIZE  
 DAP SWV8  
 LAC (I) /WORSE THAN WORST QUEUE  
 DAC SWBQ  
 SWV8, LAC . /GET CORE STATUS  
 SPQ  
 JMP SWL6 /NOBODY THERE  
 DAP SWV9  
 SWV9, LIO . /LOOK AT STATUS OF GUY THERE  
 SPI /HUNG?  
 JMP SWL6 /YES, IGNORE  
 CLA /FIGURE EFFECTIVE QUEUE

RCL 6S  
 SZA  
 CLI  
 RIR 6S  
 LAI  
 SUB SWBQ /COMPARE WITH BEST QUEUE SO FAR  
 SMA  
 JMP SWL6 /NOT AS GOOD  
 DIO SWBQ /BETTER, REMEMBER THIS GUY  
 LAC SWV8 /REMEMBER CORE  
 DAP RCORE  
 LAC SWV9 /REMEMBER GUY  
 DAP RSTAT  
 SWL6, IDX SWV8 /LOOK AT NEXT CORE  
 SAS (LAC CS3+1)  
 JMP SWV8  
 LAC SWBQ /NO MORE CORES, DID WE FIND ANYONE?  
 SAD (I)  
 JMP SWNN /NO ONE WANTS TO RUN, WAIT  
 DAP SWL28 /SAVE QUEUE OF BEST GUY  
 AND (1777)  
 SAD (1777) /X777?  
 JMP SWL26 /YES  
 SWL30, LAW I CS1-1  
 ADD RCORE /SETUP TOP OF RCORE  
 RAR 4S  
 DIP RCORE  
 SAR 2S /SETUP TO RENAME  
 DCH (ADD SWV10)

SWV10, RNM  
 LAW STAT  
 SAD RSTAT  
 JMP RUNDDT  
 EEM  
 SWV19, JMP SWL32 /SWITCH

SWL31, LAW SWL32  
DAP SWV19 /RESTORE SWITCH  
LAW 200  
XAI  
LIO C16SU  
SAD (700)  
LIO C16PSU  
DIO I (PC)  
DAC I (BILLTT)  
DZM I (TOTRT)  
DZM I (TOTRT+1)  
DZM I (IOPRT)  
DZM I (IOPRT+1)  
DZM I (BITS)  
DZM I (NOCHAR)  
LSM  
LAC I DAT  
LIO I TME  
ESM  
DAC I (SUPTD)  
DIO I (SUPTD+1)

SWL32, LEM  
LAC I RSTAT /GET GUY'S STATUS  
DAP RUNTM  
RAL 5S  
SPA  
JMP DDTWS /EXEC DDT WANTS TO SEE THIS GUY  
RAR 1S  
SPA  
JMP IOPWS1 /THE IO PROCESSOR WANTS HIM  
RAR 1S  
SPA  
JMP BRKS /HE HIT THE BREAK KEY  
EEM /HE REALLY WANTS TO RUN  
LIO I (PC)  
DIO 75 /SETUP PC IN DEBREAK AREA  
JSP C16TUS /UNSAVE CORE 16 IF NECESSARY  
LIO I (FLAGS)  
LFI  
LAC I (AC)  
DAC 74  
LIO I (IO)  
DIO 76  
JMP SWDBK /DEBREAK  
  
SWL26, LAW STAT /INIT TO CHANGE ALL X777'S TO X776  
DAP SWV18  
  
SWL28, LAW . /LAW X777  
SAS .  
JMP SWL29 /NOT X777, IGNORE  
LAW I 1  
ADD I SWV18  
DAP I SWV18 /STORE X776  
  
SWL29, IDX SWV18  
SUB (SAS STAT-WHERE)  
SAS WTOP  
JMP SWL28  
JMP SWL30

SWFC,  
SZF 4 /WHERE IS BEST GUY?  
JMP SWBDR /READ FROM FASTRAND  
LAW I WHERE-STAT /STARTUP STUFF  
ADD SWV7 /GET STAT PTR.  
DAP I SWT2 /SAY HE'S IN CORE  
DAP RSTAT /SAY HE'S RUNNING  
LAC SWT2  
DAC RCORE /SAY THIS CORE IS RUNNING  
LIO I SWV7 /GET TT #  
SUB (CS1-1-400) /GET 400 CORE PTR  
DAC I SWV7 /REPLACE IN WHERE TABLE  
LAW SWL31 /SETUP SWITCH  
DAP SWV19  
JMP SWL30 /GO RUN GUY

S WNN,  
SZS 10  
JSP CEXECCTT  
LAW 374  
AND TTF  
RAR 2S  
LIA  
LAC I TTF  
RAR 1S  
SPA  
LFI  
CLA  
SAS HOTFLG  
JMP SWNN /WAIT FOR SOMETHING TO HAPPEN  
JMP SWORG /SOMETHING HAPPENED, GO HANDLE IT

SWDBK1,  
RCK  
DIO SWAPT  
LAC 74  
LIO 76  
ASC 1700  
JMP I 75 /DEBREAK

SWAPD,  
LIO I (PRIORITY)  
CLA "U" CMI  
RCR 8S  
JMP SWAPG

/BREAK KEY HAS BEEN HIT  
BRKS, LAC RSTAT /GUY BEING FASTRAND SWAPPED?  
SAD BDSTAT  
JMP SWL33 /YES, IGNORE BREAK KEY  
LAC (-40000) /REMOVE BREAK KEY BIT  
LSM  
AND I RSTAT  
DAC I RSTAT  
ESM  
EEM  
LAC I (BITS) /IS HE RUNNING UNDER IDDT  
SMA  
JMP BRK1+1 /NO : GO CHECK RESTART MODE  
LAW 77 /WAS THE TT THE BILLTT  
AND I (BILLTT)  
JDA TTNUM /SET UP POINTERS  
LAW .+3 /USE TT ROUTINES TO CHECK  
DAP TTCKSX  
JMP TTCKS1+2  
SAS (2) /.+3 : IS IT A BREAK  
JMP BRK1 /NOT BILLTT : IS USER IN RESTART MODE  
LAC I (AC) /COPY AC,PC,IO,FLAGS  
DAC I (HH16AC)  
LIO I (IO)  
DIO I (HH16IO)  
LIO I (FLAGS)  
DIO I (HH16FG)  
LAC (-500000) /REMOVE IDDT AND RESTART BITS  
AND I (BITS)  
DAC I (BITS)  
LAC C16BKK  
LIO I (PC)  
DIO I (HH16PC)  
JMP BRK3

BRK1, LAC I (BITS) /IS USER IN RESTART MODE  
BRK1+1, AND (-100000)  
SAD I (BITS)  
JMP ESWORG /USER NOT IN RESTART MODE  
IOR (40000) /SET DEBREAK LEGAL BIT  
DAC I (BITS)  
LIO I (PC)  
DIO I (RESTPC)  
LAW RESTSU

BRK3, DAC 75 /RESET PC TO APPROPRIATE POSITION  
LAW 1 /SET TO HIGH QUE  
DAC RUNTM  
JMP SWL34

/FASTRAND SWAPPER  
SWBD, SZF I 4 /SWAP OR WRITE ONLY  
JMP SWBDW /WRITE ONLY  
LAW WHAT  
LIO SWFBDF /GET SLOT TO WRITE ON (WS)  
AAI  
DAP SWBDV1  
LAC I SWV7 /GET SLOT TO READ FROM (RS)  
DAC SWFBDF /NEW SWAPPING SLOT  
ADD (WHAT)  
DAP SWBDV2  
LAC I SWV16 /GET 40C WHERE C IS CORE INVOLVED  
DAC I SWV7 /SAY GUY ON RS IS NOW IN CORE  
ADD (CS1-1-400)  
DAP SWV4 /CS PTR TO CORE  
DIO I SWV16 /SAY GUY WHO'S IN CORE IS NOW ON WS  
LAW I STAT  
ADD I SWV4  
SWBDV1, DAC . /WHAT PTR TO WS  
SWBDL1, LAW STAT /READ ONLY COMES IN HERE  
SWBDV2, ADD . /WHAT PTR TO RS  
DAC I SWV4 /SAY CORE HAS GUY WHO USED TO BE ON RS  
LAW 201 /CODE: SAVED FOR SWAPPING  
SZF 3  
LAW 400 /READ ONLY, CODE: FREE  
DAC I SWBDV2 /SETUP NEW STATUS OF RS  
LAW I WHAT 100 /WRITE ONLY COMES IN HERE  
ADD SWBDV1 /CALCULATE WRITE SLOT  
RCR 9S  
LAW I WHAT 100  
ADD SWBDV2 /CALCULATE READ SLOT  
RCR 9S  
LAW I CS1-1  
ADD SWV4 /GET CORE  
RAR 6S /POSITION  
DCH (ADD .+1) /SETUP TO RENAME THAT CORE TO 0  
RNM  
EEM  
LAC I (AC)  
DAC I (FASTAC)  
LAC I (PC)  
DAC I (FASTPC)  
DIO I (AC)  
LAC C16FSW /STARTING ADDRESS FOR SWAP  
SZF 3  
LAC C16FIN /SA FOR READ ONLY  
SZF I 4  
LAC C16FOU /SA FOR WRITE ONLY  
DAC I (PC)  
LEM

LAW SWBDL5  
SZF I 4  
LAW SWBDL4 /SPECIAL CODING ON RETURN FROM WRITE ONLY  
DAP SWBDV4  
LAC I SWV4  
LSM  
DAC BDSTAT  
LAC I BDSTAT  
DAC FSTAT  
ESM  
LAW 1  
DAC I BDSTAT  
JMP SWORG

/WRITE ONLY ENTRY  
SWBDW, LAW CS1-1-400  
ADD I SWV16  
DAP SWV4 /SETUP CS PTR TO CORE  
LAC SWV11 /GET FREE BD FIELD FOUND  
DAP SWBDV1 /SETUP PTR  
JMP SWBDL2

/READ ONLY ENTRY  
SWBDR, LAW WHAT  
ADD I SWV7 /GET RS PTR  
DAP SWBDV2  
LAW I CS1-1-400  
ADD SWV4 /GET CORE PTR  
DAP I SWV7  
JMP SWBDL1

/ ERROR RETURN FROM FASTRAND SWAP  
SWFERR, LAW I STAT  
ADD RSTAT  
RCR 9S  
LAW I 3  
LSM  
JDA 14SOCT  
LAC (SWFERT)  
JDA 14STXT  
ESM

/ RETURN FROM FASTRAND SWAPPING (CHANNEL 16)  
SWFAST, LAC I (FASTAC)

DAC 70  
LAC I (FASTPC)  
DAC 71  
LEM  
LSM  
LAC FSTAT  
DAC I RSTAT  
DAP RUNTM  
DZM BDSTAT  
ESM

SWBDV4, JMP . /SWITCH

SWBDL4, DZM I RCORE /WRITE ONLY, MARK CORE AS FREE

LAW I STAT  
ADD RSTAT  
DAC I SWBDV1 /MARK WS AS CONTAINING GUY

ADD (WHERE)

DAP SWBDV5  
LAW I WHAT

ADD SWBDV1

SWBDV5, DAP . /MARK GUY AS BEING ON FASTRAND

SWBDL5, DZM HOTFLG /TELL SWAPPER IN ALL 3 CASES

ISB 1700

JMP R0

SWFERT: 662226  
006451  
517756

/ SUBROUTINES TO SAVE AND UNSAVE CORE 16 VARIABLES IF NECESSARY

C16TSV, DAP C16TSX  
LAW I 7777  
NAI  
RIL 2S /PC IN IO  
SPI I  
C16TSX, JMP .  
DAC C16TS1  
LAW C16TEM  
DAC C16TS2  
C16TSI, LAW I 10  
DAC C16TS3  
C16TSL, LAC I C16TS1  
DAC I C16TS2  
IDX C16TS1  
IDX C16TS2  
ISP C16TS3  
JMP C16TSL  
JMP C16TSX

C16TUS, DAP C16TSX  
LAW I 7777  
NAI  
RIL 2S  
SPI I  
JMP C16TSX  
DAC C16TS2  
LAW C16TEM  
DAC C16TS1  
JMP C16TSI

C16TS1, 0  
C16TS2, 0  
C16TS3, 0

IOPWS1, LAC (-20000) /SET HUNG BIT @ CLEAR WANTS BIT  
LSM  
AND I RSTAT  
DAC I RSTAT /AND FLUSH WANT BIT  
ESM  
LAC RSTAT  
DAC I STAT  
LAW I 1  
DAC I RCORE  
LAC RCORE  
DAC I CORE  
SAR 2S  
IOR (100000) /RENAME CORE TO 4  
DCH (ADD +1)  
RNM  
LAW I 7777  
AND ICORE /RETURN TO IOP WITH REALCORE IN AC  
EEM  
JMP I IOPWGO /GO TO CORE 17

C EXECCTT, DAP C EXECX  
LAW TTP" T" LAC  
IOR EXECTT  
SAL 2S  
DAP C EXEC1  
ADD (3)  
DAP C EXEC2  
LAW 101  
C EXEC1,  
C EXEC2,  
LAT  
AND (77)  
DAC EXECTT  
SAL 2S  
ADD (TTP)  
DAP C EXEC3  
ADD (3)  
DAP C EXEC4  
LAW 105  
C EXEC3,  
DAP •  
LAW STAT 0  
C EXEC4,  
DAP •  
C EXECX,  
JMP •

L DIA, 0  
L DWC, 0  
L DCL, 0  
L DSTAT, 0  
B DSTAT, 0  
F STAT, 0  
S WWQ, 0  
S WBQ, 0  
S WT1, 0  
S WT2, 0  
S WT3, 0  
R UNTM, 0  
HOTFLG, -0  
W TOP, WHERE+1

R CORE, 0  
I CORE, 0  
DCORE, 0  
R STAT, 0  
I STAT, 0  
T STAT, STAT+1

WHERE, 402  
REPEAT 177, 300 /UNUSED  
WHAT, 201 /L.D. FIELD 0 SAVED FOR SWAPPING  
REPEAT 37, 400 /OTHER FIELDS EMPTY  
201 /FIELD 40  
REPEAT 37, 202 /SECOND DRUM HASN'T ARRIVED  
201 /B.D. SLOT 0  
REPEAT 61, 400 /49. SLOTS  
REPEAT 16, 202 /14. MEANINGLESS PLACES

CS1, 0  
CS2, STAT 0 /XDDT IS IN CORE 10  
CS3, -3 /CORE 14 HASN7T ARRIVED YET  
SWFLDF, 0 /L.D. FIELD 0 IS SWAPPING FIELD  
40  
SWFBDF, 100

VARIABLES  
CON, CONSTANTS  
FOO,

7500/  
D SPTB, 140000 ELGIOT  
140000 DHANG  
140000 TYI  
140000 TYO  
140000 TIS  
140000 TOS  
140000 GTY  
140000 RTY  
140000 TTCKS  
140000 TTMODE  
140000 RSMC  
REPEAT 2, 140000 ELGIOT  
140000 PSU  
140000 DELAY  
REPEAT 13, 140000 ELGIOT  
140000 DWPP  
REPEAT 4, 140000 ELGIOT  
140000 DPEEK  
140000 ELGIOT  
140000 DCLR /GRDR GET THE READER  
140000 DRPA /RPA READ PUNCHED TAPE ALPHABETIC  
140000 DRLSRD /RRDR RELEASE THE READER  
140000 DGETPU /GPUN GET PUNCH  
140000 DPPA /PPA PUNCH PUNCHED TAPE ALPHABETIC  
140000 DRLSPU /RPUN RELEASE PUNCH  
140000 •RCK  
REPEAT 30, 140000 ELGIOT  
DSPTB+100,

C 16TB, REPEAT 100, 360000 10  
C 16TB+100,

C 16BKP, 360000 10  
C 16BKK, 360000 10  
C 16HH, 360000 10  
C 16HLT, 360000 10  
C 16SU, 360000 10  
C 16PSU, 360000 10  
C 16FIN, 360000 10  
C 16FOU, 360000 10  
C 16FSW, 360000 10

7714/ /CORE 15 ENTRIES

1 SECJA, DCH 1SECA  
1 SECJB, DCH 1SECB  
SORADR, DCH SORX  
1 MIADR, DCH 1MIX  
1 5SOCT, 0  
1 5S0C1, 0  
1 5STXT, 0  
1 5STX1, 0  
DAT, 0  
TME, 0  
TOTFLG, 0

740/ /CORE 17 ROUTINES

DATADR, DCH DATX  
CONADR, DCH CONX  
I OPADR, DCH IOPX  
I OPWGO, JDA

EXPUNGE TYIHNG, TYOHNG, GTYHNG, EOTHNG, RPAHNG, UNUSED, CORHNG, CLKHNG, GLPHNG

START HLT-JMP

CORE 10 ENTRIES TO CORE 14 11-10-66 (14T010,10)

100002/ DCH STAT  
DCH DDTWNT  
DCH XDDTYO  
DCH XDDTYI  
DCH DDTRPA  
DCH XRDIN  
DCH RDRLS

100015/ DCH WTOP  
DCH WHERE

START HLT-JMP

## CORE 15 ENTRIES TO CORE 14 (14T015,14)

150010/ DCH CONX  
DCH OCTR  
DCH 1MIX  
DCH RSTAT  
-STAT  
DCH R0  
DCH R1  
DCH R2  
DCH ELGIOT  
DCH DSP1  
DCH HOTFLG  
DCH C16RH  
DCH FSTAT  
DCH BDSTAT  
STAT 1  
STAT 2  
DCH SWFAST  
DCH SWFERR

START HLT-JMP

CORE 17 ENTRIES TO CORE 14 (14T017,10)

177700/ DCH DATX  
DCH IOPX  
DCH R0  
DCH R1  
DCH R2  
DCH ESWORG  
DCH HOTFLG  
DCH ISTAT  
DCH RSTAT  
DCH ICORE  
-STAT  
DCH INIT  
DSWAPX  
DCH DCORE

START CAL 400

XDDT PART 1 11-10-66 (XDDTA,13)

ASC=720051  
CAC=720053  
CBS=720056  
CKS=720033  
DSC=720050  
EEM=724074  
ESM=720055  
IOH=730000 /IN-OUT HALU  
ISB=720052  
LAT=762200  
LEM=720074  
MSM=720054  
PPA=720005  
PPB=720006  
RPA=720001  
RPB=720002  
RKB=720030  
SFT=660000  
SZS=640000  
TYI=720004  
TYO=720003  
XX=760400

DBA=722061  
DCL=720063  
DIA=720061  
DRA=722062  
DWC=720062  
ERM=720065  
LRM=720064  
RCC=723022  
RCH=722022  
RCK=720032  
RNM=720066  
RRC=722122  
RR1=720037  
RR0=720017  
RSC=721122  
RSM=720067  
RTB=720035  
SSB=724122  
TCB=724022  
TCC=725022

HALT=HLT  
TYIHNG=IOT 100  
SUPPC=40 /PMACE TO PLANT INSTRUCTION  
PC=35  
DDTCOR=100000

0/ JMP RLSE  
JMP USERR  
STAT, 0  
DWANT, 0  
DDUO, 0  
DDUI, 0  
DDTRPA, 0  
DDTCLR, 0  
DDTRRD, DDTCOR MSE  
IOPDNE, 0 /CORE 17 LINK INTO CORE 10  
12, JMP ER4 /(IT'S 20 NOW)  
DCONT, 0  
14, 0 /UNUSED  
WTOP, 0  
WHERE, 0  
17, 0 /UNUSED  
20, CLA /HALT PROCEDURE  
MSM  
SAS I DCONT  
JMP ERS /IOP BUSY  
JSP I IOPDNE  
HLT /MAY PUSH CONTINUE  
26, JMP T+1 /ENTRY FROM RESTART PROCEDURE

40/  
LOWMIM, DDTCOR+4200/

DEFINE DISP MC,UC  
Z=LC-LSE 44  
Z"T"1000 UC-LSE 44  
TERMINATE DISP

DEFINE LETTER A,B  
DISP A+MSE-44,B  
TERMINATE LETTER

DEFINE SQUEEZE A,B,C  
A"T"1600.+B"T"40.+C  
TERMINATE SQUEEZE

L=0 R=1

DDM=(DDTCOR)  
EIC=(DDTCOR+SGP)  
EVC=(DDTCOR+EST)

C20=(20)  
C77=(77)  
CI=(I)  
CJ=(JMP)  
C4=(400000)

LOW, 0 -0 /INITIAL SYMBOL TABLE  
 0 DDTCORE  
 S0OZE 0, 27, 46 MSK  
 S0OZE 0, 36, 46 EST  
 S0OZE 0, 15, 46 MEM  
 0 400000

## /OPERATE GROUP (IN EVENT TIME ORDER)

S0OZE 15, 15, 23	CCI
S0OZE 26, 13, 32	LAP
S0OZE 26, 13, 36	LAT
S0OZE 15, 26, 15	CLC
S0OZE 15, 26, 13	CLA
S0OZE 15, 26, 23	CLI
S0OZE 26, 13, 32	OPR 100 /MAP-CLA OPR
S0OZE 26, 13, 36	OPR 2000 /LAT-CLA OPR
S0OZE 15, 27, 13	CMA
S0OZE 15, 27, 23	CMI
S0OZE 22, 26, 36	HLT
S0OZE 35, 41, 32	SWP
S0OZE 26, 13, 23	LAI
S0OZE 26, 23, 13	LIA
S0OZE 35, 36, 20	STF
S0OZE 15, 26, 20	CLF
NOPCOD, S0OZE 30, 31, 32	NOP
S0OZE 31, 32, 34	OPR

## /SPECIAL OPERATE GROUP (IN EVENT TIME ORDER)

S0OZE 26, 20, 23	LFI
S0OZE 26, 23, 20	LIF
S0OZE 35, 44, 26	SZI
S0OZE 35, 15, 23	SCI
S0OZE 35, 15, 20	SCG
S0OZE 15, 26, 26	CLM
S0OZE 23, 20, 23	IFI
S0OZE 23, 23, 20	IIF
S0OZE 35, 15, 27	SCM
S0OZE 15, 27, 26	CML
S0OZE 23, 16, 13	IDA
S0OZE 23, 16, 15	IDC
S0OZE 13, 13, 23	AAI
S0OZE 23, 13, 23	IAI
S0OZE 42, 13, 23	XAI
S0OZE 35, 32, 31	SPO

## /SKIP GROUP

SQOZE 15, 26, 31	C MO
SQOZE 35, 32, 33	SPQ
SQOZE 35, 44, 27	SZM
SQOLE 35, 30, 23	SNI
SQOLE 35, 32, 23	SPI
SQOZE 35, 44, 31	SZO
SQOZE 35, 27, 13	SMA
SQOLE 35, 32, 13	SPA
SQOLE 35, 44, 13	SZA
SQOLE 35, 44, 20	SZF
SQOLE 35, 44, 35	SZS
SQOLE 35, 25, 32	SKP

## /STANDARD INSTRUCTIONS

SQOLE 0, 0, 23	I
SQOLE 13, 16, 16	ADD
SQOZE 13, 30, 16	AND
SQOZE 13, 35, 15	ASC
SQOZE 15, 13, 15	CAC
SQOZE 15, 13, 26	CAL
SQOZE 15, 14, 35	CBS
SQOZE 15, 25, 35	CKS
SQOZE 16, 13, 15	DAC
SQOZE 16, 13, 32	DAP
SQOLE 16, 14, 13	DBA
SQOLE 16, 15, 22	DCH
SQOLE 16, 15, 26	DCN
SQOLE 16, 23, 13	DIA
SQOLE 16, 23, 31	DIO
SQOZE 16, 23, 32	IIP
SQOZE 16, 23, 40	DIV
SQOZE 16, 34, 13	DRA
SQOZE 16, 35, 15	DSC
SQOZE 16, 41, 15	DWC
SQOZE 16, 44, 27	DZM
SQOZE 17, 17, 27	EEM
SQOZE 17, 34, 27	ERM
SQOZE 17, 35, 27	ESM
SQOZE 23, 16, 42	IDX
SQOZE 23, 31, 22	IOH
SQOZE 23, 31, 34	IOR
SQOZE 23, 31, 36	IOT
SQOZE 23, 35, 14	ISB
SQOZE 23, 35, 32	ISP
SQOZE 24, 16, 13	JDA
SQOZE 24, 27, 32	JMP
SQOZE 24, 35, 32	JSP
SQOZE 26, 13, 15	LAC
SQOZE 26, 13, 41	LAW
SQOZE 26, 15, 22	LCH
SQOZE 26, 17, 27	LEM
SQOZE 26, 23, 31	LIO
SQOZE 26, 34, 27	LRM

SQOZE	26, 35, 27	LSM
SQOZE	27, 37, 26	MUL
SQOZE	32, 32, 13	PPA
SQOLE	32, 32, 14	PPB
SQOZE	34, 13, 26	RAL
SQOZE	34, 13, 34	RAR
SQOZE	34, 15, 15	RCC
SQOZE	34, 15, 22	RCH
SQOZE	34, 15, 25	RCK
SQOZE	34, 15, 26	RCL
SQOZE	34, 15, 34	RCR
SQOZE	34, 23, 26	RLI
SQOZE	34, 23, 34	RIR
SQOZE	34, 30, 27	RNM
SQOZE	34, 32, 13	RPA
SQOZE	34, 32, 14	RPB
SQOZE	34, 34, 14	RRB
SQOZE	34, 34, 15	RRC
SQOZE	34, 34, 313	RRI
SQOZE	34, 34, 31	RRO
SQOZE	34, 35, 15	RSC
SQOZE	34, 35, 27	RSM
SQOZE	34, 36, 14	RTB
SQOZE	35, 13, 16	SAD
SQOZE	35, 13, 26	SAM
SQOZE	35, 13, 34	SAR
SQOZE	35, 13, 35	SAS
SQOZE	35, 15, 26	SCL
SQOZE	35, 15, 34	SCR
SQOZE	35, 20, 36	SFT
SQOZE	35, 23, 26	SIL
SQOZE	35, 23, 34	SIR
SQOZE	35, 35, 14	SSB
SQOZE	35, 37, 14	SUB
SQOZE	36, 13, 16	UAD
SQOZE	36, 15, 14	TCB
SQOZE	36, 15, 15	TCC
SQOZE	36, 43, 23	TYI
SQOZE	36, 43, 31	TYO
SQOZE	42, 15, 36	XCT
SQOZE	42, 31, 34	XOR
SQOZE	0, 42, 42	XX
SQOZE	0, 2, 35	1S
SQOZE	0, 3, 35	2S
SQOZE	0, 4, 35	3S
SQOZE	0, 5, 35	4S
SQOZE	0, 6, 35	5S
SQOZE	0, 7, 35	6S
SQOLE	0, 10, 35	7S
SQOLE	0, 11, 35	8S
SQOLE	0, 12, 35	9S

SGP,

EST, DDUCORE LOW

RLSE, JSP I DDTRRD  
LSE, JSP LCC /MISTEN LOOP  
LSE+1, CLC"U"CLF 7  
DAC SP2  
DAC OPO  
LSS, CMC  
DAC CHI  
DAC SP1  
LSP, DZM WRD  
SSN-1, LAC CUN /(IOR)  
SSN, DIP SGN  
BSLASH, DZM DOM  
DZM SYL  
N2, DZM SYM L /LOOP FOR NEXT SYLLABLE  
DZM SYM R  
DZM FSM  
CLC"U"CLF 4  
DAC LET  
DAC CC  
LSR, LIO SK1 /LOOP PER CHAR. CHEAP PLACE...  
DIO WEA /...TO DO INITIALIZATION.  
LAW VF5+1  
DAP VF5  
LAW LWU  
DAP BAX  
JSP I DDII / TYPE-IN, HANG IF 00 CHARS  
RAL 6S  
DAC CH  
CDB, LAW DTB /USED AS DTB  
DAC FM1 /NON-ZERO  
CAD, ADD CH /USED AS ADD  
DAP /+1  
LISL, MIO .  
ISP CAS  
JMP UPPER  
LSU, SZF I 4 /"QUOTE" MODE?  
JMP NOTQM /NO  
IDX CC  
SAS C4  
JMP LN4  
QUOTE2, LAC FSM  
JMP N1

## /NO-EVAL ROUTINES

ULC,       LAW I 2       /77 SEEN  
DAC CAS  
JMP MSR

EOT,       CLC           /EOT SEEN. RESET EVERYTHING FOR LARRY  
DAC MSK  
DAP UL  
DZM LL  
LAC CDCH  
DAC MEM  
DAC TAS  
LAW 8.  
DAC RADIX  
DZM USER  
DZM NUM  
LAW PI  
DAP PNS  
LAW PEV  
DAP PA1  
JMP LSE+1

CHH,       LIO LET      /NUMBER-SIGN LOOKUP  
LAC CHI  
SPI "U" SMA  
JMP SHALL  
LAW CHTBL  
SEEK,  
/AND  
YE,  
/SHALL  
/FIND  
ERR,  
XCL,  
DOT,  
QUO,

LAC .  
SZA  
JMP CHH1  
LAW CHARAC R? /PRINT Q-MARK. IGNORE LAST CHAR.  
JDA TYS  
JMP LSR

LAW XCTBL   /EXCLAMATION MARK  
JMP SEEK

LIO CC       /PERIOD  
LAC LOC  
SPI I  
LAC DNM  
DAC SYL  
DAC DNM  
LAW 44  
DAC T2  
JMP LN1

LAC FSM      // MEANS TAKE AS INTERNAL  
SZA  
JMP N1  
STF 4  
JMP LSR

Q, LAC LWT // "Q" IS LAST QUANTITY TYPED  
JMP N1

DAQ, LAW 7777 // <DEFINES SYM AS ADDRESS OF Q  
AND LWT  
JMP COM+1

COM, LAC LOC // COMMA DEFINES SYM AS LOC  
COM+1, DAC DF1  
DZM FL1

DEF, LAC LET // DEFINE SYMBOL  
SK1, SZA  
JMP ERR  
JSP DE  
JMP PN2

DEL, LAW 3 // #,DELETE  
JDA TYS  
DEL1, JMP PN2 // END OF NO EVAL ROUTINES

/SET OUTPUT RADIX

RADX, SNI"U"SZM // SET RADIX  
SAD C1  
JMP ERR  
DAC RADIX  
JMP LSE

/ARITHMETIC FUNCTIONS

PLS, LAC CAD // PLUS SIGN  
JMP SSN

MIN, SPI // MINUS SIGN  
DIO WRD  
MAC CSU // (SUB)  
JMP SSN

ISC, LAC CAN // (AND) LOGICAL "AND".  
JMP SSN

UNI, JMP SSN-1 // LOGICAL "OR"

/PRINT "Q" IN FORM SPECIFIED

EQL,        DAC LWT        /PRINT INTEGER IN CURRENT RADIX  
              JSP LCT  
              JDA OPT  
PN2,        JSP LCT  
              JMP LSS

DEC,        DAC LWT        /PRINT AS DECIMAL INTEGER  
              LAC RADIX  
              DAC T  
              LAW 10.  
              DAC RADIX  
              JSP LCT  
              JEA OPT  
              LAC T  
              DAC RADIX  
              KMP PN2

ARW,        DAC LWT        /PRINT AS IOSTRUCTION  
              JSP MCT  
              DAC PI        /JDA PI, BUT DON'T SET FLAG 2  
              LAW PN2  
              STF 3        /FORCE SYMBOLIC PRINTOUT  
              JMP PI+2

PBX,        DAC LWT        /PRINT AS INTERNAL  
              JSP LCU  
              JDA TYSA  
              JMP PN2

SQP,        DAC LWT        /PRINT AS SQUEEZE CODE  
              JSP LCT  
              DAC SYM R  
              DZM SYM L  
              LAW PN2  
              DAP SPX  
              JMP SPTMIM

/SET CURRENT CORE

CORE,       SNI I  
              JMP ERR  
              RAR 6S  
              AND C170000  
              DIP MEM  
DIP TAS      /YES, VIRGINIA, IT'S NECESSARY  
              DZM LOC  
              JMP MSE

## /REGISTER EXAMINATION AND CHANGE

VBR, AND (177777 / :  
DIP TAS  
SPI  
JMP VBR1  
DIP MEM  
JMP TAS

VBR1, AND (170000  
SZA  
BRING1 /CALL BRING WITHOUT CHANGING MEM  
JMP VBR1X

BAC, LAW OPT /OPEN BRACKET (BAR-CONSTANT)  
JMP BAS+1

BAI, LAW TYS /CONTROL "I" (BAR-INTERNAL)  
JMP BAS+1

BAS, LAW PI /CLOSED BRACKET (BAR-SYMBOLIC)  
BAS+1, DAP BAX  
VBR1X, MAC OPT  
BAR, SPI  
JMP TA6  
LIO SP1  
DIO SP2  
SPI  
JMP TA5  
AND (177777)

TR3, DAC SP2  
SP5, JSP LCT  
LAC I SP2  
JMP TA7

CR, DAC LWT /LINE FEED  
JDA MRF  
LAW 7715  
JDA TYS  
JMP MSE+1

UC8, JDA MRF /> MEANS MAKE CORR. AND OPEN REGISTER  
JMP TA6

BS,	JDA MRF LIO SP2 SPI I JMP BS2 IDX LOC BS+5, DAC LWT JMP TA3+2	/BACKSPACE
FS,	JDA MRF JSP LCC LAW I 1 LIO SP2 SPI I JMP FS1 ADD MOC JMP BS+5	/ARROW UP (FORWARD SPACE)
TAB, TA3,	JDA MRF DAC NWT JSP MCC JDA PAD	/MEMORY FIELD SWITCH
TA3+2,	LAW CHARAC R/ JDA TYS	
TA5,	DIM LOC DAP LOC LIO MEM EIO TAS	
TA6,	MIO C4 DIO SP2 DAP TAS JSP LCT BRING MAC I UAS	
TA7, BAX,	EAC LWT JDA . DZM OPN JMP PN2	/PI, OPT OR MWT

## /SYMBOL ROUTINES

VAL,        DAC DF1        /OPEN PAREN, SETS UP VALUE FOR DEFINITION  
              JMP LSS

TBL,        JSP I DDTCLR    /READ SYMBOL PUNCH  
              JMP RDNY  
              DZM FL1  
              JSP SOI  
MR1,        JSP GWD  
              DAC SYM L  
              JSP GWD  
              AND C177777  
              DAC SYM R  
              IOR SYM L  
              SZA  
              JMP TBM1  
TBN,        JSP LCT  
              LAC EST  
              JDA OPT

TBM,        KSP RBK        /SKIPS REST OF TAPE  
              JMP TBM

KIL,        SPI        /KILL SYMBOL(S)  
              JMP KI3

KI2,        LIO LEU  
              SPI  
              JMP ERR

KI1,        LAC I EV2    /USED AS LAC I  
              IOR KI1  
              DIP I EV2  
              JMP LSE

/ZERO CORE

PUL, SPI /LOWER LIMIT SETUP  
JMP ERR  
DAC FA  
JMP LSS

ZRO, JSP OK /ZERO CORE TO CONTENTS OF M#+3  
LAW 7777  
SPI  
DAC WRE  
DIP WRD  
AND FA  
SPI  
CLA  
IOR MEM  
DAC T  
LIO NUM  
BRING  
ZR2, SUB MEM  
SUB WRD  
SZM  
JMP LSE  
DIO III  
IDX T  
JMP ZR2

## /SEARCHES

EAS,	LAW EA1 JMP WS	/EFFECTIVE ADDRESS SEARCH
NWS,	LAC SK2 DAC WEA	/NOT WORD SEARCH
WDS, WS,	LAW WS1 SPI JMP ERR DAP WS2 JSP MCC LAC USER DAC MWSU DZM MWSXU DZM MWSFUI SZA I SAS MEM KMP OQJ MWS LIO MWSXU IDX USER SAD C200 JMP LOOK1 ADD STAT DAC EAS1 LAW I 7777 AND I EAS1 SAD C760000 JMP WS3+5	/WORD SEARCH  /MOBY WORD SEARCH: IF USER=MEM=0, SEARCH... /...ALL USERS FOR WRD/  /RESTORE USER; EXIT TO LSE.
NOTMWS,	BRING LAC LM IOR MEM DAC I	
WS4,	DIP T2 DZM SYM DAP T3	
WS2,	LAC I T2 JMP .	/EA1 OR WS1

## /ROUTINES COCERNED WITH TIME SHARING

LOOK, SPI /SET USER BEIOG EXAMINED  
JMP CREATE /GO CREATE NEW USER  
LIA  
SUB (82. /WOULD BE 200, BUT TOP OF...  
/STAT + WHERE TBMS SOMETIMES PATCH AREA  
SMA  
JMP ERR  
LAC STAT  
AAI  
DAC T  
LAW I 7777  
AND I T  
SAD (760000  
JMP USERR  
MOOK1, DIO USER  
JMP MSE

HOLD, SPI I  
JMP HALTI /HALT USER  
LAW (IOR (LAC)) /PREVENT USER FROM HALTING  
JMP HOLD1

FREE, MAW (AND (-LAC)) /ALMOW USER TO HALT  
HOLD1, DAP HOLD2 /SET OR CMEAR HELD BY XDDT BIT  
JSP NOUUD  
SPI I  
JMP ERR /WORKS FOR CURRENT USER ONLY  
BRING1  
LAC I C4 /(400000)  
HOLD2, XCT .  
DAC I C4  
JMP LSE

SETPTR, SPI I /FREE TT PTRS  
SPA  
JMP ERR  
SAL 2S  
ADD (141000  
DAC T  
SUB (141000+100"T"4) /100 TT'S  
SMA  
JMP ERR  
MAW 70"T"4  
AND T  
SZA I /SET UP TT BUFFER PTRS  
LAW I 100 /0-7~100  
SAS (10"T"4)  
SAD (20"T"4)  
LAW I 300 /10-27~400  
SMA  
LAW I 1100 /30-77~1400  
CMA  
LIO OPT  
SIL 3S  
ADD (140000  
AAI  
LIA  
LAW 101  
LSM  
DAC I T  
IDX T  
DIO I T  
IDX T  
DIO I T  
IDX T  
LAC I T  
DZM I T  
ESM  
JDA OPT  
JMP LSE

BGN,        BRING1        /START USER  
          SPI  
          JMP BGN3        /NO ARG  
          AND C170000  
          SZA I  
          JMP BGN1        /CORE 0  
          SZS I 60  
          JMP ER4  
BGN1,      JSP NOTU0  
          LIO OPT  
BGN2,      DIO I (PC     /ENTRY FROM "STRTUP"  
BGN3,      LAC STAT  
          ADD USER  
          DAC T  
          LAW 1  
          DAP I T        /HIGH QUEUE  
          JMP LSE  
  
XEC,      SNI I 60        /EXECUTE ARG  
          JMP ER4        /SS6 NOT UP  
          DAC .+1  
T,          0  
          NOP  
          JMP LSE  
  
/READ OR VERIFY PAPER TAPE  
  
SVFY,     LAW VF2        /SPECIAL VERIFY  
          DAP VF5  
VFY,      SPI I        /VERIFY CORE AGAINST BINARY TAPE  
          JMP ERR  
          JSP I DDTCLR  
          JMP RDNY  
          JSP LCC  
          LAC RB2        /(SAD I)  
          JMP RD1

REPEAT 1IF VP .-LSE+44-1000, PRINT .MYSTIC 1000 EXCEEDED.

RD, SPI I /READ BINARY TAPE INTO CORE  
JMP ERR  
JSP OK  
JSP I DDTCLR  
JMP RDNY  
LAC BS1 /\*(DAC I)  
RD1, DIP VF4  
JSP SOI  
BRING  
VF1, LAC MEM  
DIP T  
LAC T  
SUB LL  
SUB MEM  
SPA  
JMP VF2  
ADD LL  
SUB UL  
SZM  
JMP VF2  
LAC I LA  
VF4, T /\*DAC I OR SAD I  
JMP VF2  
SZA I  
JMP VF5 /\*SPECIAL VERIFY, IGNORE ZERO ON TAPE  
LAW I 7777  
AND I T  
ADD T  
SUB MEM  
XOR I LA  
VF5, SZA I /\*SPECIAL VERIFY, IGNORE ".." IN ADDRESS PART  
JMP . /\*JMP .+1 OR JMP VF2  
JSP PAC  
JSP LCT  
LAC I LA  
JDA LWT  
JSP LCC  
VF2, IDX T  
IDX LA  
SAD RB1  
JSP RBK  
JMP VF1

/BRANCHES. PLACED HERE TO SAVE ROOM IN DISPATCH AREA.

/REST OF LISTEN LOOP

NOTQM, CLA  
RCL 9S  
DAC T2  
SUB C44  
SPA  
JMP LN  
ADD TLS / (JMP LSE)  
DAP LSX  
SUB (JMP DEL1 / LAST NO-EVAL ROUTINE  
SPQ  
JMP LSX  
LAW SYL  
LIO LET  
SPI I / SKIP IF LETTER NOT SEEN  
JSP EVL  
JMP EV4  
ER2, LAW CHARAC RU / IGNORE INPUT AND START OVER  
JDA TYS  
JMP LSS  
EV4, DAP SGN  
LAC WRD  
SGN, XX / OPERATOR AND SYLLABLE ADDR.  
DAC WRD  
LIO CHI  
SPI  
LAC LWT  
DAC OPT  
LSX, JMP . / I. O. MINUS IF NO ARG. ARG IN AC.  
UPPER, JSP TYPOUT  
XCT LSL  
RIL 9S  
JMP LSU  
LN, ADD C44-12 / LETTER-NUMBER LOGIC  
SPA  
JMP N  
LN1, DZM LET  
DZM CHI  
IDX T2  
IDX CC  
SAS C4  
JMP LN3  
LIO SYM R  
DIO SYM L  
DZM SYM R

LN3, SUB C6  
SZM  
JMP LSR  
LAC SYM R  
RAL 2S  
ADD SYM R  
RAL 3S  
ADD T2  
DAC SYM R  
LAW I 4  
ADD CC  
SMA  
JMP LSR  
LAC FSM /ENTRY FROM QUOTE  
RAL 6S  
ADD CH  
DAC FSM  
LAW 77  
SAD CH  
JMP ULC  
LAW 3  
SZF 4  
SAS CC  
JMP LSR  
JMP QUOTE2

N, LAC SYL  
RAL 3S  
CUN, IOR T2 /USED AS IOR  
DAC SYL  
LAC DNM  
RAL 2S  
ADD DNM  
RAL 1S  
ADD T2  
DAC DNM  
JMP LN1

## /REST OF NUMBER-SIGN LOOKUP

SHALL, LAC SYL  
 SPA  
 JMP ERR  
 SAL 2S  
 SUB (100" T" 4)  
 SMA  
 JMP ERR  
 ADD (100" T" 4 + 141000  
 JMP DDS

CHH1, SAD SYM R  
 JMP FIND  
 IDX YE  
 IDX YE  
 JMP YE

FIND, IDX YE  
 DAP .+1  
 JMP .

STATUS, LAC STAT  
 ADD USER  
 JMP DDS

C, LAW MEM  
 JMP F+1

M, LAW MSK  
 JMP F+1

F, LAW EST  
 F+1, IOR DDM  
 DDS, DAC SP1  
 N1, DZM CHI  
 DAC SYL  
 DAC DNM  
 JMP N2

## /MODE CHANGING ROUTINES

LOT, LAW TYSA  
 JMP CNS+1

SMB, LAW PI  
 JMP CNS+1

CNS, LAW OPT /SYMBOLIC-CONSTANT-FLEXO SWITCH SETUP  
 CNS+1, DAP PNS  
 JMP LSE

OAD, LAW PVM  
 JMP RAD+1

RAD, LAW PEV /OCTAL-RELATIVE SWITCH SETUP  
 RAD+1, DAP PA1  
 TLS, JMP LSE /USED AS LSE

/REST OF BS

BS2, IDX SP2  
SP3, DAC LWT  
DIP MEM  
JDA PAD  
LAW CHARAC R/  
JDA TYS  
LAC TAS  
DIP MEM  
KMP SP5

/REST OF FS

FS1, ADD SP2  
DAP SP2  
JMP SP3

/REST OF KIL

KI3, LAC CDDTCOR LOW  
DAC EST  
JMP LSE

/REST OF TBL

TBL1, JSP GWD  
DAC DF1  
LAC SYM L  
LIO SYM L  
RIL IS  
SMA"U"SPI  
JMP MR1  
AND C177777  
DAC SYM L  
JSP DE  
JMP MR1

/REST OF EFFECTIVE ADDRESS SEARCH

EA1, DAC EA1 /SAVE INSTRUCTION  
LIA  
AND C770000  
SAD CCAL  
JMP EA4  
SAD CJDA  
JMP EA6  
SAD CLAW I  
JMP WS3  
AND C760000  
SAS COPR /FLUSH THESE  
SAD CSPO  
JMP WS3

SAS (SKP  
SAD (SFT  
JMP WS3  
SAD (IOT  
JMP WS3  
SAS (DCH  
SAD (LCH  
JMP EA2  
LAI  
EA3., AND CI /IS THERE AN I BIT  
SK1., SZA  
JMP EA2 /YES  
LAC EAS1  
AND (760000  
SAD (XCT)  
JMP EA11  
SAS (LCH  
SAD (DCH  
JMP EA5  
WS8, LAW 7777 /NO, AND OFF INSTRUCTION PART OF FIRST WORD  
AND I T2  
  
WS1, XOR WRD /COMPARE  
CAN, AND MSK /USED AS AND  
WEA, XX /SZA OR SZA I  
JMP WS3 /SETUP NEXT WORD  
  
WS6, LAC MWSU  
SAD MEM  
SAS MWSFTI  
JMP NOTMW1  
IDX MWSFTI  
JSP LCC  
LAC USER  
DAC MWSXU  
JDA OPT  
LAC (FLEXO "L"  
JDA TYS  
JSP LCC  
NOTMW1, LAW LCC  
  
PAC, DAP PAX /HIT, PRINT OUT  
LAC T  
DAP LOC  
JDA PAD  
LAW CHARAC R/  
JDA TYS  
JSP LCT  
LAC I T  
JDA LWT  
JSP .

WS3,       IDX T           /INDEX  
SUB MEM  
SUB UL  
SPQ  
JMP WSNW  
WS3+5,      LAC MWSU  
SAS MEM  
JMP LSE  
JMP MWSNU

WSNW,       LAC T  
JMP WS4

EA4,        LAC WRD       /CAL FINDS 100,101, AND ITS ADDRESS  
SAS (100  
SAD (101  
JMP WS6  
JMP WS8      /CHECK ADDRESS

EA6,        JSP EAS1+1     /JDA,CHECK THING ADDRESSED  
IDX EAS1     /CHECK ADDRESS+1

EA7,        AND (7777  
JMP WS1

EA2,        LAW EA3.       /GET REG. REFERENCED FOR I,LCH,DCH  
JMP EA12

EA11,       LAW EA1  
EA12,       DAP EA13  
LAI  
JDA EAS1  
IDX SYM  
SAD (10.  
JMP WS3  
LAI  
DAP T2  
LAC I T2  
LIA  
JMP .

EA13,       LAI           /LCH,DCH PTR  
LIO EAS1     /INSTR.  
RIL 5S  
SPI  
IDC  
JDA EAS1     /CHECK LOC. REFERENCED BY POINTER  
LAC T2  
JMP EA7      /CHECK LOC OF POINTER

EAS1,       0           /SPECIAL COMPARE  
EAS1+1,      DAP EAS2  
LAW 7777  
AND EAS1  
XOR WRD  
AND MSK  
SZA

EAS2,       JMP .       /MISS,RETURN  
JMP WS6      /HIT,PRINT AND QUIT

/REST OF "LOOK"; CREATE NEW USER.

CREATE, LIO C500  
LAC C140000  
LSM  
IOR I WTOP  
DAC T  
SUB WHERE  
DAC OPT  
SUB C82. /TOP OF STAT + WHERE TBLS PATCH AREA  
SMA  
JMP ER5  
IDX I WTOP  
DIO I T  
ESM  
LAC OPT  
DAC USER  
JDA OPT  
BRIOG1  
LAW 11  
DAC T  
MIO NUM  
STRTU1, DIO I T  
IDX T  
SAS CI  
JMP STRTU1  
LAC CTYIHOG  
STRU UP, BRING1  
DEM I C4 /CLEAR "BITS" IN CASE HALTIOG  
DAC I CSUPPC  
LIO CSUPPC  
JMP BGN2

/REST OF "HOLD"; HALT CURRENT USER.

HALT1, SAS C1  
JMP ERR  
JSP NOTU0  
LAC CHALT  
JMP STRTUP

START

XDDT PART 2 11-10-66 (XDDTB,11)

## /SUBROUTINES

MRF, 0 /PUT AC IN CURRENTLY OPEN REG...  
DAP MRX /...IF THERE IS ONE.  
LAC OPN  
SMA  
SPI  
JMP MRF1  
MAC SP2  
LIO MRF  
SMA  
JMP SPECIAL  
JSP OK  
BRING  
DIO I TAS  
MRF1, LAC MRF  
MRX, JMP .

SPECIAL, SAD (DDTCOR+MEM  
JMP ER4 /TRIED TO CHANGE C#  
DIO I SP2  
JMP MRF1

## /SYMBOLS

DE, DAP DEX /DEFINE SYMBOL  
JSP EVL  
JMP DF2  
DE1, LAC (DDTCOR+LOWLIM+6)  
SUB EST  
SMA  
JMP DEX /NOP TO READ SYMS BELOW LOWLIM  
IDX EST  
LIO FM1  
LAC MEM  
SNI I  
LAC C4  
DAC TYS  
LIO DF1  
SAD I EST  
JMP DE2  
LAW I 2  
ADD EST  
DAC EST  
DE2, DIO I EST  
LAW I 1  
ADD EST  
DAC EST  
LIO SYM R  
DIO I EST  
SUB ONE  
DAC EST  
LAC SYM L  
SZA I  
JMP DE3  
IOR C4  
DAC I EST  
LAW I 1

DE3, ADD EST  
DAC PAD  
SUB ONE  
DAC EST  
DZM I EST  
LAC TYS  
BS1, DAC I PAD /USED AS DAC I  
JMP DEX

DF2, LAC DF1  
SAD I ES4  
DEX, JMP .  
DAC I ES4 /VALUE IS CHANGING  
LAC EV2  
SUB (DDTCOR+LOW  
SPA  
JMP DEX  
LAC I EV2 /KILL INITIAL SYMBOL  
IOR K11 /(LAC I)  
DIP I EV2  
JMP DE1 /DEFINE SYMBOL

EVL, DAP EVX /EVALUATE SYMBOL  
LAC EST  
DAC ES4  
EVG, DAC EV2  
LAC I ES4  
SPA  
JMP ESN  
SZA I  
JMP EV5  
RAL 1S  
SPA  
JMP ESI  
CLA  
ES3, SAS SYM L  
JMP ESI  
LAC SYM R  
RB2, SAD I ES4 /USED AS SAD I  
JMP EV3  
ESI, IDX ES4  
EV6, IDX ES4  
SAS EVC  
JMP EVG  
EV8, IDX EVX  
EV3, IDX ES4  
EVX, JMP .

ESN, IDX ES4  
LAC I EV2  
RAL 1S  
SPA  
JMP ESI  
SAR 1S  
JMP ES3

EV5, IDX ES4  
LAC I ES4  
SMA  
SAD MEM  
JMP EV6  
IDX ES4  
EV7, LAC I ES4  
SZA I  
JMP EV5  
SPA  
IDX ES4  
IDX ES4  
IDX ES4  
SAS EVC  
JMP EV7  
JMP EV8

PI, XX /PRINT INSTRUACION  
STF 2  
PI+2, DAP PX+1  
JSP PEV  
LAC PI  
SUB CI  
SPA  
JMP PPK  
DAC PI  
PR1, LAW 0  
JDA TOU  
MAW CHARAC RI  
JDA TYS  
PPK, CLA  
JDA TOU  
XCT EA+1  
JMP PVL  
LAC LWT  
AND C760000  
SAD CSFT  
JMP I66  
SAD PR1 /LAW 0  
JMP PLO  
RAR 1S  
SZA  
CSU, SUB C320000 /USED AS SUB  
SPA  
JMP PLO  
PVL, LAC PI  
PVL+1, SZA I  
SZF 1 I  
PV3, JDA OPT  
PX, CLF 7 /EXIT  
PX+1, JMP .

PLO, JSP PEV  
JMP PA1+1

I66, LAW 1 /1S-9S  
ADD PI  
AND PI  
SZA  
JMP PVL  
LAW PA1+1  
DAP PEX  
LAC EIC  
JMP EA1+2

PAD, 0 /PRINT ADDRESS  
DAP PX+1  
LAW 7777  
AND PAD  
DAC PI  
CLF 1  
PA1, JSP PEV /PEV OR PVL  
PA1+1, LAW CHARAC R+  
JDA TYS  
JMP PVL

PEV, DAP PEX /SYMBOL LOOKUP SUBR  
LAW I 7777  
AND PI  
SAD COPR /DETECT OPERATES  
JMP I76  
AND C760000  
SAS CSKP  
SAD CSP0  
JMP SEV

EAK, DAP EA+1  
LAC EST  
EAK+2, DAC ES4  
CLF 1  
EAL, LAC I ES4  
LIO I ES4  
SZA I  
JMP EI1  
RAL 1S  
SPA  
JMP EI3  
LAC ES4  
SPI  
IDX ES4  
DAC OP1  
SPI I  
CLI  
DIO T3  
IDX ES4

EA, LAC I ES4  
EA+1, SKP I  
JMP SKO  
XOR PI  
SPA  
JMP EIX  
LAC PI  
SUB I ES4  
SPA  
JMP EIX  
SZF I 1  
JMP PSW  
LAC I ES4  
SUB I OP2  
SZM  
JMP PSW  
EIX, IDX ES4 /LOOK AT NEXT SYMBOL  
SAD EVC /END OF TABLE  
JMP EIX+5  
SAS EIC  
JMP EAL  
EIX+5, XCT EA+1  
JMP PEX  
SZF I 1  
JMP PVL /TYPE OUT REST IN OCTAL  
LAC PI  
SUB I OP2  
LIA  
SZA /DETECT NEG NUMS  
JMP I77  
EIY-1, DIO PI  
EIY, JSP SPT /PRINT SYMBOL  
LAC PI  
SK2, SZA I  
JMP PX  
XCT EA+1  
JMP .+2  
PEX, JMP .  
CMA  
DAC T22 /MASK  
JMP EIX  
  
EI1, IDX ES4 /STUFF FOR SYMBOL SEARCH  
LAC I ES4  
SMA  
SAD MEM  
JMP EIX  
IDX ES4  
EI2, LAC I ES4  
SZA I  
JMP EI1  
SPA  
IDX ES4  
IDX ES4  
IDX ES4  
SAS EIC  
JMP EI2  
JMP EIX+3

E13, SPI  
IDX ES4  
IDX ES4  
JMP EIX

I76, DAC T1  
LAC PI  
SAS (NOP  
JMP SEV+2  
CLA  
DAP EA+1  
LAC (DDTCORE+NOPCOD  
JMP EAK+2

SEV, DAC T1 /SAVE INSTRUCTION  
LAC PI  
SEV+2, CMA  
DAC T22 /MASK  
LAW 600 /SPA"U"SMA-SKP  
JMP EAK

SKO, IOR T1  
SAS I ES4  
JMP EIX  
SZF 1  
XOR T1  
SZA I  
JMP EIX  
XOR PI  
LIA  
AND T22  
SZA  
JMP EIX  
DIO PI  
LAC (FLEXO "U"  
SZF 1  
JDA TYS  
PSW, LAC I OP1  
DAC SYM R  
LIO T3  
DIO SYM L  
LAC ES4  
DAC OP2  
STF 1  
XCT EA+1  
JMP EIY  
JMP EIX

I77, LAW I 7777  
AND PI  
SAS (770000  
JMP EIY-1  
LAW CHARAC R-  
JDA TYS  
LAC PI  
CMA  
JMP PV3

SPT, DAP SPX /SYMBOL PRINT SUBROUTINE  
LAW 7777  
AND I OP2  
SAS I OP2  
JMP SPTMIM /INST.  
ADD PI  
SZF I 2  
JMP SPT1  
SAD LOC  
JMP PRNTPT  
SPT1, LIA  
SUB I OP2  
SUB (100 /LIMIT TAGS TO (VALUE)+100  
SMA  
SZF 3  
JMP SPTLIM  
LAI  
JMP PVL+1 /PRINT AS OCTAL  
SPTLIM, LAC CDDTCORE SYM L  
DAC T3  
SPB, LAW SPD  
DAP SPJ  
SPN, DZM OP1  
SPR, IDX OP1  
SPV, LAC I T3  
AND (177777  
SPJ, SUB .  
SPA  
JMP SPP  
SPU, DAC I T3  
JMP SPR  
PRNTPT, DZM PI  
DEM OP2  
LAW CHARAC R.  
JDA TYS  
JMP SPX  
SPP, LAC OP1  
SCR 1S  
SZA I  
JMP SPS  
ADD SPT / (SPX)  
DAP .+1  
LAC .  
SPI I  
RAR 6S  
JDA TOU  
SPS, IDX SPJ  
SAS (SUB SPD+3  
JMP SPN  
IDX T3  
SAS CDDTCORE SYM+2  
JMP SPB  
SPX, JMP .

/UNSQUOZE TABLE. MUST FOLLOW SPX.

SPL,	FLEXO 01	FLEXO 23
	FLEXO 45	FLEXO 67
	FLEXO 89	FLEXO AB
	FMEXO CD	FLEXO EF
	FLEXO GH	FLEXO IJ
	FLEXO KL	FLEXO MN
	FLEXO OP	FLEXO QR
	FLEXO ST	FLEXO UV
	FLEXO WX	FLEXO YZ
	1603	/FLEXO •#

SPD,	3100
	50
ONE,	1

/TYPEOUT SUBROUTINES

MCC,	DAP LC1	/LOWER-CASE-CARRIAGE-RETURN
	LAW 76	/TYPE CRLF
MC1,	JDA TYS	
	JMP .	
LCT,	DAP LCX	/LOWER-CASE-TAB
	CLA	
	JDA TYSA	
MCX,	JMP .	
LWT,	0	/LAST WORD TYPED
	DAP PNX	
	LAC LWT	
PNS,	JDA PI	/PI, OPT, OR TYSA
PNX,	JMP .	
TYS,	0	/TYPE SYMBOL, ETC.
TYS+2,	DZM TSWICH	
	DAP TYX	
	LAW I 3	
	DAC TYSVAR	
TYL,	LAC TYS	
	RAL 6S	
	DAC TYS	
	AND C77	
	DAC CH	/SAVE CHAR
	SAD TSWICH	/IF 0, DON'T TYPE SPACES
	JMP TYL1	
	SAD C77	/IF 77, TYPE "CHAR"
	JMP TYSA1	
	JDA TOU	
TYL1,	ISP TYSVAR	
	JMP TYL	
	LAC LWT	
	JMP .	

TYSA, 0 /TYPE 3 CHARS  
DAC TSWICH  
LIO TYSA  
DIO TYS  
JMP TYS+2

TYSA1, LAC TYS  
RAL 6S  
DAC TYS  
AND C77  
DAC CH  
ISP TYSVAR /TYPOUT UNLESS 77 IS THIRD BYTE  
JSP TYPOUT  
SAS CH  
JMP TYL1  
RAL 6S  
IOR C77  
JMP TYPOU2

TYPOUT, DAP TYPEX /TYPE CONTROL CHARACTERS  
LAC CH  
SAS (12  
SAD (15  
JMP TYPEX  
SAD (4  
JMP TYPEX  
IOR (40  
SAD CH  
JMP TYPEX  
RAL 6S  
IOR (FLEXO " ")  
JDA TOU  
RAR 6S  
JDA TOU  
RAR 6S  
JDA TOU  
JMP . /AC HAS CH IF NOTHING HAPPENED

OPT,	0	/ANY RADIX PRINT
	DAP OPX	
	DZM OP1	
OPA-1,	LAC OPT	
OPA,	DAC OP2	
	CLI"U"SWP	
	RCL 1S	
	DIV RADIX	
OP1,	0	
	SAS OP1	
	JMP OPA	
	SWP	
	ADD C20	
	JDA TOU	
	LAC OP2	
	DAC OP1	
	SAS OPT	
	JMP OPA-1	
	LAW 10.	
	SAS RADIX	
OPX,	JMP .	
	LAW CHARAC R.	
	JDA TYS	
	LAC OPT	/IS THIS NEEDED?
	JMP OPX	
TOU,	0	/TYPE CHAR FROM BOTTOM OF AC
	DAP TOUX	
	LIO TOU	
	JSP I DDTO	/TYPE FROM BOTTOM OF I. O.
TOUX,	LAC TOU	
	JMP .	

## /PAPER TAPE SUBROUTINES

SO1,	JDA RDB	
SOI,	JDA RDB	/ SKIP OVER INPUT ROUTINE
	SPI I	
	JMP SO1	
RBK,	DAP RBX	/READ A BLOCK INTO BUFFER
	JSP RDB+1	
RB3,	LAW BUF	
	DAP RB1	
	DAP LA	
	LAC DDM	
	DIP LA	
	DZM CHI	/CKSUM
	DIO T2	
	DIO T	
	SPI	
	JMP RLSE	
	JSP RDB+1	
	DIO CH	
	LAW I 1	
	ADD CH	
	SUB T2	
	AND C-77	
	SZA	
	JMP RBX+1	

RB0, JSP RDB+1  
DIO I RB1  
LAC I RB1  
ADD CHI  
DAC CHI  
IDX RB1  
IDX T2  
SAS CH  
JMP RB0  
ADD CHI  
ADD T  
JDA RDB  
SAD RD4 /WORD JUST READ IS IN RD4  
RBX, JMP .  
RBX+1, LAC (FLEXO SUM /CHECKSUM ERROR  
ER3, JDA TYS  
JMP RLSE

GWD, DAP GWX /GET WORD FROM READER BUFFER  
GWD+1, LAC LA  
SAS RB1  
JMP GWD1  
JSP RBK  
JMP GWD+1  
GWD1, DAP GWD2  
IDX LA  
GWD2, LAC .  
GWX, JMP .

RDB, 0 / RPB SUBROUTINE  
RDB+1, DAP RDX  
LAW I 3  
DAC RD6  
RD3, JSP I DDTRPA  
RIR 8S  
SPI I  
JMP RD3  
RIL 2S  
LAC RD4  
RCL 6S  
DAC RD4  
ISP RD6  
JMP RD3  
LIO RD4  
LAC RDB  
RDX, JMP .

/TIME SHARING SUBROUTINES

OK, DAP OKX /SS6 PROTECT EXEC AND USER ZERO  
CLA  
SAD MEM  
JSP NOTU0  
SAS MEM  
SZS 60  
OKX, JMP .  
ER5, ESM  
ER4, LAW CHARAC R?  
JMP ER3

NOTU0, DAP NOTU0X  
CLA  
SAS USER  
NOTU0X, JMP .  
JMP ER4

BRING=JDA .  
WANTED, 0 /GET USER INTO CORE ZERO  
DAP WANTX  
LAC MEM  
SZA  
WANT1, JMP WANTX-1  
DIO SAVEIO  
LIO USER  
SNI I /0" L" FOR NO BRIOG  
JSP I DWANT / USER WANTED IN CORE ZERO  
LIO SAVEIO  
WANTX-1, LAC WANTED  
WANTX, JMP .

BRING1=JDA .  
WANT2, 0 /BRING REGARDLESS OF VALUE IN MEM  
DAP WANTX  
LAC WANT2  
DAC WANTED  
JMP WANT1

USERR, LAW FLEXO NO  
JDA TYS  
CLA  
JDA TOU  
LAW FLEXO US  
JDA TYS  
LAW FLEXO ER  
JMP ER3

RDNY, LAW FLEXO BU  
JDA TYS  
LAW FLEXO SY  
JMP ER3

CHtbl, IRP [X,,T,M,S,C],[Y,,F,M,STATUS,C]  
CHARAC R'X-26  
JMP Y  
ENDIRP  
0

XCTBL, IRP [X,,C,A,R,S,I],[Y,,CNS,OAD,RAD,SMB,LOT]  
CHARAC R'X-26  
JMP Y  
ENDIRP  
0

DTB, DISP PLS,LSE  
DISP XCL,ISC  
DISP QUO,SVFY  
DISP CHH,CORE  
DISP ERR,EOT  
DISP ERR,EAS  
DISP ISC,FREE  
DISP ERR,BGN  
DISP VAL,HOLD  
DISP DEF,BAI  
DISP DEC,CR  
DISP PLS,KIL  
DISP COM,LOOK  
DISP MIN,ERR  
DISP DOT,NWS  
DISP BAR,ERR  
LETTER 0,SETPTR  
LETTER 1,0  
LETTER 2,RADX  
LETTER 3,SQP  
LETTER 4,TBL  
LETTER 5,UNI  
LETTER 6,VFY  
LETTER 7,WDS  
METTER 8,XEC  
LETTER 9,RD  
DISP VBR,ZRO  
DISP PUL,ERR  
DISP DAQ,ERR  
DISP EQL,ERR  
DISP UC8,ERR  
DISP PBX,ERR  
DTB 40, DISP ERR,ERR  
REPEAT 3,LETTER .-DTB-27,ERR  
LETTER .-DTB-27,BSLASH  
LETTER .-DTB-27,ERR  
LETTER .-DTB-27,FS  
LETTER .-DTB-27,ARW  
REPEAT 23,LETTER .-DTB-27,ERR  
DISP BAC,ERR  
DISP TAB,DEL  
DISP BAS,ERR  
DISP BS,ERR  
DISP ULC,ERR  
DTB+100,

## "/"PERMANENT" VARIABLES

MSK,	-0	/FIRST 4 STAY TOGETHER. MASK FOR WS.
LL,	0	/LOWER LIMIT FOR WS AND RD
UL,	7777	/UPPER LIMIT FOR WS AND RD
NUM,	0	/VALUE STORED BY ZRO
USER,	1	
RADIX,	8.	
FA,	0	/FIRST ADR FOR "Z". SET AT PUL BY SEMICOMON.
MEM,	DDTCOR	/CURRENT CORE
LOC,	0	/CURRENT LOCATION

## /OTHERS

WRD,	0	/INPUT WORD FOR LSE
SYL,	0	/OCTAL VERSION OF INPUT SYLLABLE FOR LSE
DNM,	0	/DECIMAL DITTO
FSM,	0	/FLEXO DITTO
CAS,	0	/CASE FOR LSE. SET TO -2 WHEN 77 CH SEEN.
CC,	0	/NUMBER OF CHARS THIS SYLLABLE FOR LSE
CHI,	0	/+0 IF ANY CHARS THIS WORD FOR LSE; ...
		/...-0 IF NONE. ALSO TEM (CKSUM) FOR RBK.
LET,	0	/+0 IF LETTER THIS SYL FOR LSE; -0 IF OOT.
CH,	0	/CHAR FOR LSE, TYS AND TYSA, AND TYPOUT...
		/...TEM (END TEST) FOR RBK.
SYM,	0	/L HALF OF SYMBOL FOR LSE, TBL, DE, PEV,...
		/...SPT, EVL. TEM (DEPTH OF TRACE) FOR WS.
SYM+1,	0	/R HALF
SP1,	0	/SET IN LSE. -0 IF NO #-SIGN THIS WORD...
		/...VALUE IF #-SIGN SEEN.
SP2,	0	/SET BY LSE AND REGISTER EXAMINING ROUTINES...
		/...ADR OF OPEN #-SIGN REG. MINUS IF NONE.
OPN,	0	/0 IF REG OPEN; -0 IF NOT...
		/...SET AT LSE AND BAX.
TAS,	DDTCOR	/ADR OF OPEN REG. SET BY REG EXAMINING...
		/...ROUTIOES. USED BY MRF IN CLOSING REG.

FL1, 0 /ZERO IF SYM TO BE DEFINED BY DE AS LOCAL;...  
DF1, 0 /...NON-ZERO, GLOBAL. SET AT LSE, COM, TBL/  
EV2, 0 /VALUE OF SYM TO BE DEFINED BY DE...  
ES4, 0 /...SET AT COM, VAL, TBL.  
T1, 0 /SET BY EVL. POINTS AT FISST WORD OF...  
U3, 0 /...SYMBOL. USED BY KIL.  
OP2, 0 /SET BY EVL. POINTS AT VAL OF SYM. USED...  
RB1, DDTCOR+BUF /...BY DE. AMSO SYM VAL PTR FOR PEV.  
T22, 0 /TEM (INSTR) FOR PEV  
T2, 0 /TEM (MASK) FOR PEV  
U3, 0 /TEM FOR PEV AND SPT  
RD6, 0 /TEM (VALUE) FOR PEV AND SPT. TEM FOR OPT.  
LA, 0 /READER BUFFER INPUT PTR...  
RD4, 0 /...USED BY VFY, RBK, GWD/  
TYSVAR, 0 /READER BUFFER OUTPUT PTR  
TSWICH, 0 /BINARY WORD BEIOG BUILT BY RDB  
/ISP COUNTER FOR RDB  
/COUNTER FOR TYS AND TYSA  
/FLAG FOR TYS AND TYSA...  
/...IF ZERO, DON'T TYPE SPACES.  
T2, 0 /TEM FOR LSE, WS, RBK  
SAVEIO, 0 /SAVED I. O. FOR BRING  
MWSU, 0 /USER ON ENTRY TO MOBY WORD SEARCH  
MWSXU, 0 /USER ON EXIT FROM MOBY WORD SEARCH  
MWSFTI, 0 /FIRST TIME INDICATOR FOR MOBY WS

CON, CONSTANUS

BUF, .+100/

FOO, FLEXO FOO

REPEAT 0IF P,[  
PRINT /FOO+1/  
-[RD-MSE+44-1000]/ PRINT /SPACE LEFT IN DISPATCH AREA/  
]

REPEAT 1IF P,EXPUNGE L,R

START HLT-JMP