

PAGE 1

XDDT PART 2 6-30-67 (XDDTB, 12)

/SUBROUTINES

```
MRF,      0          /PUT AC IN CURRENTLY OPEN REG...
           DAP MRX    /...IF THERE IS ONE.
           LAC OPN
           SMA
           SPI
           JMP MRF1
           LAC 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 FL1
           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 KI1        /(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	
EV7,	IDX ES4	
	LAC I ES4	
	SZA I	
	JMP EV5	
	SPA	
	IDX ES4	
	IDX ES4	
	IDX ES4	
	SAS EVC	
	JMP EV7	
	JMP EV8	
PI,	XX	/PRINT INSTRUCTION
	STF 2	
PI+2,	DAP PX+1	
	JSP PEV	
	LAC PI	
	SUB CI	
	SPA	
	JMP PPK	
	DAC PI	
PR1,	LAW 0	
	JDA TOU	
	LAW CHARAC RI	
	JDA TYS	
PPK,	CLA	
	JDA TOU	
	XCT EA+1	
	JMP PVL	
	LAC LWT	
	AND (760000)	
	SAD (SFT	
	JMP I66	
	SAD PR1	/LAW 0
	JMP PLO	
	RAR 1S	
	SZA	
CSU,	SUB (320000	/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	/S-9S
	ADD PI	
	AND PI	
	SZA	
	JMP PVL	
	LAW PA1+1	
	DAP PEX	
	LAC EIC	
	JMP EAK+2	
PAD,	Ø	/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 (OPR	/DETECT OPERATES
	JMP I76	
	AND (760000	
	SAS (SKP	
	SAD (SPO	
	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	

JHP BIX+3

EI3, 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↑USMA-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 SPTLIM       /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 (DDTCORE 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
          DZM 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 (DDTCORE SYM+2
          JMP SPB
SPX,      JMP .

```

/UNSQOZE TABLE. MUST FOLLOW SPX.

SPL,	FLEXO 01	FLEXO 23
	FLEXO 45	FLEXO 67
	FLEXO 89	FLEXO AB
	FLEXO 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

LCC,	DAP LC1	/LOWER-CASE-CARRIAGE-RETURN
	LAW 76	/TYPE CRLF
	JDA TYS	

LC1, JMP .

LCT,	DAP LCX	/LOWER-CASE-TAB
	CLA	
	JDA TYSA	

LCX, JMP .

LWT,	0	/LAST WORD TYPED
	DAP PNX	
	LAC LWT	

PNS,	JDA PI	/PI, OPT, OR TYSA
PNX,	JMP .	

TYS,	0	/TYPE SYMBOL, ETC.
	DZM TSWICH	

TYS+2, 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

TYX, 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      /TYP OUT UNLESS 77 IS THIRD BYTE
            JSP TYP OUT
            SAS CH
            JMP TYL1
            RAL 6S
            IOR C77
            JMP TYP OU2

TYP OUT,   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
TYP OU2,   JDA TOU
            RAR 6S
            JDA TOU
TYPEX,     JMP .      /AC HAS CH IF NOTHING HAPPENED

```

```

OPT,      Ø           /ANY RADIX PRINT
          DAP OPX
          DZM OP1
OPA-1,    LAC OPT
OPA,      DAC OP2
          CLI↑USWP
          RCL 1S
          DIV RADIX
OP1,      Ø
          SAS OP1
          JMP OPA
          SWP
          ADD (2Ø
          JDA TOU
          LAC OP2
          DAC OP1
          SAS OPT
          JMP OPA-1
          LAW 1Ø.
          SAS RADIX
OPX,      JMP .
          LAW CHARAC R.
          JDA TYS
          LAC OPT      /IS THIS NEEDED?
          JMP OPX

TOU,      Ø           /TYPE CHAR FROM BOTTOM OF AC
          DAP TOUX
          LIO TOU
          JSP I DDTO    /TYPE FROM BOTTOM OF I. O.
          LAC TOU
TOUX,     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 (-77

```

SZA
JMP RBX+1

```

RBØ,      JSP RDB+1
          DIO I RB1
          LAC I RB1
          ADD CHI
          DAC CHI
          IDX RB1
          IDX T2
          SAS CH
          JMP RBØ
          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,      Ø           / 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 NOTUØ
          SAS MEM
          SZS 6Ø
OKX,      JMP .
ER5,      ESM

```

ER4, . LAW CHARAC R?
JMP ER3

```

NOTUØ,    DAP NOTUØX
           CLA
           SAS USER
NOTUØX,    JMP .
           JMP ER4

```

```

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

```

```

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

```

```

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

```

```

RDNY,      LAW FLE XO BU
           JDA TYS
           LAW FLE XO SY
           JMP ER3

```

```

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

```

```

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

```


DTB, DISP PLS,LSE
 DISP XCL,SVFY
 DISP QUO,ERR
 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,Q
 LETTER 2,RADX
 LETTER 3,SQP
 LETTER 4,TBL
 LETTER 5,UNI
 LETTER 6,VFY
 LETTER 7,WDS
 LETTER 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 ZERO
USER,	1	
RADIX,	8.	
FA,	0	/FIRST ADR FOR "Z". SET AT PUL BY SEMICOLON.
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 NOT.
CH,	0	/CHAR FOR LSE, TYS AND TYSA, AND TYP0UT...
		/...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...
		/...ROUTINES. USED BY MRF IN CLOSING REG.

```

FL1,      0      /ZERO IF SYM TO BE DEFINED BY DE AS LOCAL;...
              /...NON-ZERO, GLOBAL. SET AT LSE, COM, TBL.
DF1,      0      /VALUE OF SYM TO BE DEFINED BY DE...
              /...SET AT COM, VAL, TBL.
EV2,      0      /SET BY EVL. POINTS AT FIRST WORD OF...
              /...SYMBOL. USED BY KIL.
ES4,      0      /SET BY EVL. POINTS AT VAL OF SYM. USED...
              /...BY DE. ALSO SYM VAL PTR FOR PEV.
T1,       0      /TEM (INSTR) FOR PEV
T22,      0      /TEM (MASK) FOR PEV
T3,       0      /TEM FOR PEV AND SPT
OP2,      0      /TEM (VALUE) FOR PEV AND SPT. TEM FOR OPT.
RB1,      DDTCOR+BUF /READER BUFFER INPUT PTR...
              /...USED BY VFY, RBK, GWD.
LA,       0      /READER BUFFER OUTPUT PTR
RD4,      0      /BINARY WORD BEING BUILT BY RDB
RD6,      0      /ISP COUNTER FOR RDB
TYSVAR,   0      /COUNTER FOR TYS AND TISA
TSWICH,   0      /FLAG FOR TYS AND TISA...
              /...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,      CONSTANTS

BUF,      DDTCOR+BUF+100/

FOO,      FLEXO FOO

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

REPEAT 1IF P,EXPUNGE L,R

START HLT-JMP

```