RCSL No: 991 10172

Edition: August 1985

Author: Lars Bone Jørgensen

Title:

CPU823
Microprogram

RCSL Nr. 46-F 0087

A/S REGNECENTRALEN
af 1979

Keywords:

RC8000, CPU823, RC8006-MP, Listing.

Abstract:

Listing of the microprogram and additional tables for CPU823.

The CPU823 is used for RC8000 in connection with Multi Processor configuration.

(266 printed pages)

RCSL Nr. 46-F 0087

Printed by A/S Regnecentralen af 1979, Copenhagen

INTRODUCTION

This manual is the listings of the micro program for the CPU823, which is the CPU used in MP configurations of RC8000. This manual gives only a survey of the micro instructions, the different formats and the layout (page 2). A detailed description can be found in the Technical manual for the CPU821.

The manual consists of 5 parts:

1. Microinstruction format, definition of microinstruction parameters and Microinstructions.

2. CPU823 Microprogram listing.

3. Address calculation decoding map.

4. Instruction decoding map.

5. Prefetch and operandfetch decoding map.

The microprogram is written in symbolic assembly language using the RC3600 macro assembler DOMAC, and additional programs for formatting of listings and for generating the PROM's.

The microinstruction definitions consists of, the definitions of parameters and comments for the instruction definitions.

The microprogram listing consists of 5 columns with the following information.

1. Line number, 2 decimal digits
2. Control store address, 5 octal digits
3. Contents of control store location, 16 octal digits
4. Microprogram instruction
5. Comments preceded by a semicolon

FORMAT 0: Read/Load Scratchpad

0	1–4	5	6–7	9–10	19–20	24–25	29–30	35–36	41–42	47
P	NEXT	CL	F = 0,0,0	ALU: D / OP / FUNC / C	A	B	UNIT FUNC/ TEST	SCRATCHP. ADDR.	INT/EXT REG DEST	

FORMAT 1: Read/Load Register

0	1–4	5	6–7	9–10	19–20	24–25	29–30	35–36	41–42	47
P	NEXT	CL	F = 0,0,1	ALU: D / OP / FUNC / C	A	B	UNIT FUNC/ TEST	INT REG SOURCE	INT/EXT REG DEST	

FORMAT 2: Read Constant

0	1–4	5	6–7	9–10	19–20	24–25	29–30	35–36	41–42	47
P	NEXT	CL	F = 0,1,0	ALU: D / OP / FUNC / C	A	B	UNIT FUNC/ TEST	CONSTANT ADDR	INT/EXT REG DEST	

FORMAT 3: Read External Register

0	1–4	5	6–7	9–10	19–20	24–25	29–30	35–36	41–42	47
P	NEXT	CL	F = 0,1,1	ALU: D / OP / FUNC / C	A	B	UNIT FUNC/ TEST	EXT REG SOURCE	INT/EXT REG DEST	

FORMAT 4: Half-Word Manipulator

0	1–4	5	6–7	9–10	19–20	24–25	29–30	35–36	41–42	47
P	NEXT	CL	F = 1,0,0	ALU: D / OP / FUNC / C	A	B	NOT USED	MANIPULATOR FUNCTION	NOT USED	

FORMAT 5: Load Immediate

0	1–4	5	6–7	9–10	19–20	43–44	47
P	NEXT	CL	F = 1,0,1	ALU: D / OP / FUNC / C	24-BIT IMMEDIATE OPERAND		NOT USED

FORMAT 6: Conditional Jump

0	1–4	5	6–7	9–10	19–20	24–25	27–28	29	30	31	35–36	47
P	NEXT	CL	F = 1,1,0	ALU: D / OP / FUNC / C	A	NOT USED	S	X	T	COND SELECT	JUMP ADDRESS	

FORMAT 7: Shift, Multiply, Divide

0	1–4	5	6–7	9–10	19–20	24–25	29–30	31	35–36	37–38	40–41	42–43	44–46	47
P	NEXT	CL	F = 1,1,1	ALU: D / OP / FUNC / C	A	B	T	COND SELECT	NS	SI		M,D	TST	NOT USED

Figure 1: Microinstruction formats.

```
 0001 .MAIN DOMUS MACRO ASSEMBLER REV 02.00
01
02                ;FILE: TMAC3, EDITED: 850328/LBJ
03
04                ; CPU820   DEFINITION OF MICROINSTRUCTIONS
05                ; AND MICROINSTRUCTION PARAMETERS
06                .XPNG
07          000001 .NOMA 1
08          000000 LIST=0  ; 0: MACROS ARE NOT LISTED
09
10                ;*****************************************
11                ;DEFINITION OF MICROINSTRUCTION PARAMETER
12                ;*****************************************
13
14                ;ALU FUNCTION AND CARRY, MIR(15:19)
15
16          000000 ADD=0    ; R+S
17          000001 ADDO=1   ; R+S+1
18          000002 ADDC=2   ; R+S+C
19          000003 ADDX=3   ; R+S+ADDCOND
20          000004 SUN=4    ; -R+S-1
21          000005 SUNO=5   ; -R+S
22          000006 SUNC=6   ; -R+S-1+C
23          000007 SUNX=7   ; -R+S-1+ADDCOND
24          000010 SUB=10   ; R-S-1
25          000011 SUBO=11  ; R-S
26          000012 SUBC=12  ; R-S-1+C
27          000013 SUBX=13  ; R-S-1+ADDCOND
28          000014 OR=14    ; R OR S
29          000020 AND=20   ; R AND S
30          000024 CAND=24  ; ,-R AND S
31          000030 EXOR=30  ; R EXOR S
32          000034 EXNOR=34; ,-(R EXOR S)
33
```

```
10002 .MAIN
01
02              ;GENERAL REGISTER ADDRESSSES, MIR(20:24) AND MIR(25:29)
03
04        000000 W=0      ; W-FIELD OF INSTR. REGISTER
05        000001 WPRE=1   ; WPRE OF INSTR. REGISTER
06        000002 X=2      ; X-FIELD OF INSTRUCTION REGISTER
07        000003 LAX=3    ; LAST(21:22)
08        000020 W0=20    ; W0
09        000021 W1=21    ; W1
10        000022 W2=22    ; W2
11        000023 W3=23    ; W3
12        000024 STAT=24  ; CPU STATUS
13        000025 CAUSE=25; CAUSE REGISTER
14        000026 SB=26    ; SB REGISTER
15        000027 INF=27   ; INF REGISTER
16        000030 .2=30    ; 2
17        000031 CNTR=31  ; CONTROL OUTPUT COPY REGISTER
18        000032 WRKC=32  ; WORK REGISTER
19        000033 WRK1=33  ; DO.
20        000034 WRK2=34  ; DO.
21        000035 WRK3=35  ; DO.
22        000036 WRK4=36  ; DO.
23        000037 WRK5=37  ; DO.
24
25              ;SCRATCHPAD ADDRESSES, MIR(36:41)
26
27        000000 CPAC=0   ; CPA LIMIT COPY
28        000001 BASEC=1  ; BASE COPY
29        000002 LLIMC=2  ; LOWER LIMIT COPY
30        000003 ULIMC=3  ; UPPER LIMIT COPY
31        000004 ILIMC=4  ; INTERRUPT LIMIT COPY
32        000005 SIZE=5   ; SIZE REGISTER
33        000006 MONT=6   ; MONITOR TOP REGISTER
34        000007 CUR=7    ; CURRENT PROCESS REGISTER
35        000010 PUIX=10  ; PU INDEX
36        000011 PUTAB=11; PU TABLE
37        000012 EXOFS=12; OFFSET IN PROC. DESC. OF EXCEPTION ADDRESS
38        000013 DMOFS=13; OFFSET IN PROC. DESC. OF DUMP AREA
39        000014 SWRKC=14; WRK REGISTER 0
40        000015 SWRK1=15;
41        000016 DATSW=16; DATA SWITCH
42        000017 REGSW=17; REGISTER SWITCH
```

```
100C3 .MAIN
01
02                ;CONSTANT ADDRESSES, MIR(36:41)
03        000035 .1B0=35 ; BIT 0
04        000033 .1B1=33 ; BIT 1
05        000032 .1B2=32 ; BIT 2
06        000031 .1B3=31 ; BIT 3
07        000027 .1B4=27 ; BIT 4
08        000026 .1B5=26 ; BIT 5
09        000025 .1B6=25 ; BIT 6
10        000024 .1B7=24 ; BIT 7
11        000023 .1B[illegible]23; BIT 12
12        000022 .1[illegible]22; BIT 14
13        000021 .1[illegible]21; BIT 16
14        000020 .1B[illegible]20; BIT 17
15        000017 .1B18[illegible]17; BIT 18
16        000016 .1B19=16; BIT 19
17        000036 .1B20=36; BIT 20
18        000015 .1B21=15; BIT 21
19        000037 .1B23=37; BIT 23
20        000037 .1=37    ; 1
21        000014 .3=14    ; 3
22        000015 .4=15    ; 4
23        000030 .6=30    ; 6
24        000034 .7=34    ; 7
25        000036 .8=36    ; 8
26        000013 .10=13   ; 10
27        000012 .11=12   ; 11
28        000011 .12=11   ; 12
29        000010 .14=10   ; 14
30        000007 .15=7    ; 15
31        000016 .16=16   ; 16
32        000006 .21=6    ; 21
33        000005 .22=5    ; 22
34        000004 .23=4    ; 23
35        000017 .32=17   ; 32
36        000002 .38=2    ; 38
37        000020 .64=20   ; 64
38        000003 B12=3    ; BIT(12:23)
39        000001 B1219=1  ; BIT(12:19)
40        000000 B1214=0  ; BIT(12:14)
```

```
10004 .MAIN
01
02              ;SOURCE REGISTER ADDRESSES, MIR(36:41)
03
04              ;INTERNAL REGISTERS
05
06        000000 DISP=0   ; INSTRUCTION DISPLACEMENT, D
07        000010 ICS=10   ; INSTRUCTION COUNTER, SOURCE
08        000020 DIC=20   ; DISP+IC
09        000030 LCS=30   ; LOOP COUNTER, SOURCE
10        000040 STOC=40  ; DBUS(0:21) CON OVERFLOW CON CARRY
11        000041 STSZ=41  ; DBUS(0:21) CON SHIFT OFL. CON ZERO
12        000042 STZZ=42  ; DBUS(0:21) CON ZERO CON ZERO
13        000050 SWICH=50; SWITCHES AND CONFIGURATION
14
15              ;EXTERNAL REGISTERS
16
17        000000 CDATI=0  ; CPU DATAIN
18        000001 CIOST=1  ; CPU I/O STATUS
19        000002 RTC=2    ; REAL TIME CLOCK
20        000003 ILEV=3   ; INTERRUPT LEVEL
21        000004 TPIN=4   ; TECHNITIANS CONSOLE, DATAIN
22        000005 TCPST=5  ; TECHNITIANS CONSOLE, STATUS
23
24        000040 DATI=40  ; CACHE/CPU DATAIN
25        000041 IOSTA=41; CACHE/CPU I/O STATUS
26
27        000070 FPUR0=70; FPU FRACTION(0:23)
28        000071 FPUR1=71; FPU FRACTION(24:35) CON EXP(0:11)
29        000072 FPUST=72; FPU EXCEPTION(22:23)
```

```
10005 .MAIN
01              ;DESTINATION REGISTER ADDRESSES, MIR(42:47)
02
03                      ; 0=NO LOAD
04      000001 LAST=1   ; LOGICAL ADDRESS.STATUS
05                      ; 2= SCRATCH PAD
06      000003 ICD=3    ; INSTRUCTION COUNTER, DESTINATION
07      000004 LCD=4    ; LOOP COUNTER, DESTINATION
08      000005 CPUST=5  ; CPU STATUS REGISTER
09      000006 CNTRO=6  ; CONTROL OUTPUT REGISTER
10                      ; NOT USED
11                      ; NOT USER
12      000011 LAD=11   ; LOGICAL ADDRESS & LOG ADDR.STATUS
13      000012 DATO=12  ; DATA OUT
14      000013 BASE=13  ; BASE
15      000014 LLIM=14  ; LOWER LIMIT
16      000015 ULIM=15  ; UPPER LIMIT
17      000016 CPA=16   ; CPA LIMIT
18      000017 RTCR=17  ; REAL TIME CLOCK, READ COMMAND
19      000020 INTRG=20; CLEAR ADDRESSED INTERRUPT LEVEL
20      000021 ILIM=21  ; INTERRUPT LIMIT
21      000022 ILEVR=22; INTERRUPT LEVEL, READ COMMAND
22      000023 TPOUT=23; TECHNICIANS CONSOLE, DATAOUT
23      000024 CAMBC=24; CACHE BYPASS CONTROL
24      000025 TCCLR=25; CLEAR TC INTERRUPT
25
26      000040 CAMTD=40; CACHE TEST DATA
27      000041 CAMCN=41; CACHE CONTROL
28
29      000070 FPOP0=70; FPU FRACTION(0:23)
30      000071 FPOP1=71; FPU FRACTION(24:35) CON EXP(0:11)
```

```
10006 .MAIN
01
02              ;JUMP CONDITIONS, MIR(31:35)
03
04        000000 FALSE=0 ; THIS CONDITION IS ALWAYS FALSE
05        000001 NEG=1   ; RESULT<0 , F(0)
06        000002 NZ=2    ; RESULT<>0
07        000003 OVFL=3  ; ARITHMETIC OVERFLOW
08        000004 CARRY=4 ; ARITHMETIC CARRY
09        000005 SIGN=5  ; ARITHMETIC SIGN, F(0) EXOR OVERFLOW
10        000006 NORM0=6 ; RES(0)<>RES(1)
11        000007 NORM1=7 ; RES(1)<>RES(2)
12        000010 DIVOF=10; DIVISION OVERFLOW, LINK<>CARRY(0)
13        000011 LCN=11  ; LC(0)
14        000012 MAXLC=12; LC=0 OR LC>48
15        000013 NWADR=13; NOT W-ADDRESS, LAST(0:20)<>0
16        000014 NGADR=14; LA(0)
17                       ;
18        000016 MMODE=16; MONITOR MODE
19        000017 EMODE=17; ESCAPE MODE
20        000020 CPUIN=20; CPU INTERRUPT
21        000021 EXINT=21; DEVICE INTERRUPT
22        000022 NTCPI=22; NOT TCP INPUT
23        000023 NAUXI=23; NOT AUX INTERRUPT
24        000024 NCAME=24; NOT CAM ERROR
25        000025 NOCPA=25; NOT OCP AUTOLOAD
26        000026 NREMA=26; NOT REMOTE AUTOLOAD
27        000027 PLOW=27 ; POWER LOW
28
29              ;JUMP CONDITION CONTROL, MIR(30)
30
31        000000 F=0     ; THE COMPLEMENTED VALUE OF THE COND. IS SELECTED
32        000001 T=1     ; THE TRUE VALUE OF THE COND. IS SELECTED
33
34              ;SHIFT-IN CONTROL, MIR(37:38)
35
36        000000 Z=0     ; ZERO
37        000001 LNK=1   ; SHIFT LINK
38        000002 ADC=2   ; ADDCOND, LEFT SHIFTS ONLY
39        000002 F0=2    ; F(0), RIGHT SHIFTS ONLY
40        000003 SGN=3   ; SIGN EXTENSION, RIGHT SHIFTS ONLY
```

```
10007 .MAIN
01
02              ;TEST CONTROL, MIR(42:43)
03
04        000000 DVS=0    ; DIVSIGN:=F(0)
05        000001 DC=1     ; ADDCOND:=DIVIDE CONDITION
06        000002 MC=2     ; ADDCOND:=MULTIPLY CONDITION
07        000003 NL=3     ; NO LOAD
08
09              ;SEQUENCE CONTROL, MIR(1:4)
10
11        000003 CONT=3   ; CONTINUE
12        000000 POP=0    ; POP AND CONTINUE
13        000001 PUSH=1   ; PUSH AND CONTINUE
14        000010 RTN=10   ; SUBROUTINE RETURN
15        000013 LRTN=13  ; LOOP RETURN, CONDITIONAL
16        000016 NEXT=16  ; GOTO ADDRESS CALCULATION
17        000017 EXEC=17  ; GOTO INSTRUCTION EXECUTION
18
19              ;TEST FUNCTIONS, MIR(30:35)
20
21        000060 TSTIN=60; INTERNAL INTERRUPT + DEV. INTERRUPT
22        000050 TSTBE=50; I/O ERROR
23        000041 TSTLI=41; LIMIT VIOLATION
24        000051 TSTIO=51; I/O ERROR + LIMIT VIOLATION
25        000044 TSTOV=44; INTEGER MASK & OVERFLOW
26        000042 TSTSH=42; INTEGER MASK & SHIFT OVERFLOW
27        000071 TIOIN=71; I/O ERROR+ LIMIT VIOLATION + INTERRUPT
28        000064 TOVIN=64; INTEGER MASK & OVERFLOW + INTERRUPT
29        000062 TSHIN=62; INTEGER MASK & SHIFT OVERFLOW + INTERRUPT
30        000061 TLIIN=61; INTERRUPT + LIMIT VIOLATION
```

```
10008 .MAIN
01
02              ;UNIT FUNCTIONS, MIR(30:35)
03
04        000021 READ=21 ; READ, NO LIMIT CHECK
05        000020 READP=20; READ, W. LIMIT CHECK
06        000023 WRT=23  ; WRITE, NO LIMIT CHECK
07        000022 WRTP=22 ; WRITE, W. LIMIT CHECK
08        000010 READC=10; READ W. PROTECTION IF MEM. REF. INSTRUCTION
09        000024 READJ=24; READ W. PROTECTION, FETCHED OP. IS USED AS INSTRUCTION
10        000025 REAJ=25 ; READ NO PROTECTION, FETCHED OP IS USED AS INSTRUCTION
11                       ; UNIT FUNCTIONS FOR FPU:
12        000000 FAOP=0  ; FLOATING ADD
13        000001 FSOP=1  ; FLOATING SUBTRACT
14        000002 FMOP=2  ; FLOATING MULTIPLY
15        000003 FDOP=3  ; FLOATING DIVIDE
16        000004 WMOP=4  ; INTEGER WORD MULTIPLY
17
18              ;HALF-WORD MANIPULATOR FUNCTIONS, MIR(36:41)
19
20        000000 HLOAD=0 ;  LOAD HALF WORD
21                       ; ODD=0: SBUS:= 12 EXT 0 CON DBUS(0:11)
22                       ; ODD=1: SBUS:=12 EXT 0 CON DBUS(12:23)
23        000001 EXLOA=1 ;  LOAD EXTENDED
24                       ; ODD=0: SBUS:=12 EXT DBUS(0) CON DBUS(0:11)
25                       ; ODD=1: SBUS:=12 EXT DBUS(12) CON DBUS(12:23)
26        000002 HSTOR=2 ;  STORE HALF WORD
27                       ; ODD=0; SBUS:= DBUS(12:23) CON 12 EXT 0
28                       ; ODD=1; SBUS:= 12 EXT 0 CON DBUS(12:23)
29        000003 MASK=3  ;  SET MASK
30                       ; ODD=0; SBUS:= 12 EXT 0 CON 12 EXT 1
31                       ; ODD=1; SBUS:= 12 EXT 1 CON 12 EXT 0
32        000004 EXT=4   ;  EXTEND
33                       ; SBUS:=12 EXT DBUS(12) CON DBUS(12:23)
34        000005 SWAP=5  ;  SWAP
35                       ; SBUS:= DBUS(12:23) CON DBUS(0:11)
36        000006 LSWAP=6 ;  SWAP LEFT HALF WORD
37                       ; SBUS:=12 EXT 0 CON DBUS(0:11)
38        000007 RSWAP=7 ;  SWAP RIGHT HALF WORD
39                       ; SBUS:=DBUS(12:23) CON 12 EXT 0
```

```
10009 .MAIN
01
02                ;VALUE OF EXTERNAL ADDRESSES
03
04          000060 IDLE=20*3
05          000071 INTIN=23*3
06          000113 ESCAP=31*3
07          000116 INTC=32*3
08          000121 INTB1=33*3
09          000124 SYFL1=34*3
10          000127 INTF=35*3
11          000132 PFINT=36*3
12          000135 LOCKM=37*3
13          006000 CHCLK=2000*3
14          006003 DISPA=2001*3
15          014000 POWUP=4000*3
16          014003 TCPIN=4001*3
17          014006 OCPAL=4002*3
18          014011 REMAL=4003*3
19          014014 SINGL=4004*3
20          014017 TCPER=4005*3
21          014022 TPOCT=4006*3
22          014025 CRLF=4007*3
23          014030 GCHAR=4010*3
24          014033 TCPI2=4011*3
25          014036 DELAY=4012*3
26          014041 TPTXT=4013*3
27          014044 GTNUM=4014*3
28          014047 RAUTO=4015*3
29          014052 CINTR=4016*3
30          014055 CAMTEST=4017*3
31          014060 CAMER=4020*3
32          014063 SYFLT=4021*3
33          022000 TX=6000*3
34          022003 ATEST=6001*3
35          022006 TERROR=6002*3
36          022011 LOOP=6003*3
37          022014 SIZEM=6004*3
38          022017 TO.4=6005*3
```

```
!0011 .MAIN
01              ;*********************************
02              ; DEFINITIONS OF MICROINSTRUCTIONS
03              ;*********************************
04
05
06
07              ;************************************************************
08              ;FORMAT 0: ARITHMETRIC-LOGIC INSTRUCTIONS WITH SCRATCHPAD
09              ; SOURCE/DESTINATION.
10              ;************************************************************
11
12              ;------------------------------------------------------------
13              ;LOAD SCRATCHPAD
14
15              ;       AQS     ALUFUNC,GRA,SP                  NEXT UF/TEST CL
16              ;       ABS     ALUFUNC,GRA,GRB,SP              NEXT UF/TEST CL
17              ;       ZQS     ALUFUNC,SP                      NEXT UF/TEST CL
18              ;       ZBS     ALUFUNC,GRB,SP                  NEXT UF/TEST CL
19              ;       ZAS     ALUFUNC,GRA,SP                  NEXT UF/TEST CL
20
21              ;------------------------------------------------------------
22
```

```
10012 .MAIN
01              ; INSTRUCTIONS WITH SCRATCHPAD SOURCE
02              ;=====================================
03
04              ;-------------------------------------------------------------
05              ;NO LOAD
06
07              ;       SA      ALUFUNC,SP,GRA                  NEXT UF/TEST CL
08              ;       SQ      ALUFUNC,SP                      NEXT UF/TEST CL
09              ;       SZ      ALUFUNC,SP                      NEXT UF/TEST CL
10
11              ;-------------------------------------------------------------
12
13              ;-------------------------------------------------------------
14              ;LOAD GRB
15
16              ;       SAB     ALUFUNC,SP,GRA,GRB              NEXT UF/TEST CL
17              ;       SQB     ALUFUNC,SP,GRB                  NEXT UF/TEST CL
18              ;       SZB     ALUFUNC,SP,GRB                  NEXT UF/TEST CL
19
20              ;-------------------------------------------------------------
21
```

```
10013 .MAIN
01              ;-------------------------------------------------------
02              ;LOAD Q
03
04              ;       SAQ     ALUFUNC,SP,GRA                  NEXT UF/TEST CL
05              ;       SQQ     ALUFUNC,SP                      NEXT UF/TEST CL
06              ;       SZQ     ALUFUNC,SP                      NEXT UF/TEST CL
07
08              ;-------------------------------------------------------
09
10
11              ;-------------------------------------------------------
12              ;LOAD DESTINATIONS REGISTER
13
14              ;       SAR     ALUFUNC,SP,GRA,DR               NEXT UF/TEST CL
15              ;       SQR     ALUFUNC,SP,DR                   NEXT UF/TEST CL
16              ;       SZR     ALUFUNC,SP,DR                   NEXT UF/TEST CL
17
18              ;-------------------------------------------------------
19
10014 .MAIN
01
02              ;-------------------------------------------------------
03              ;LOAD GRB, DESTINATIONS REGISTER
04
05              ;       SABR    ALUFUNC,SP,GRA,GRB,DR           NEXT UF/TEST CL
06              ;       SQBR    ALUFUNC,SP,GRB,DR               NEXT UF/TEST CL
07              ;       SZBR    ALUFUNC,SP,GRB,DR               NEXT UF/TEST CL
08
09              ;-------------------------------------------------------
10
11
12              ;-------------------------------------------------------
13              ;LOAD Q, DESTINATION REGISTER
14
15              ;       SAQR    ALUFUNC,SP,GRA,DR               NEXT UF/TEST CL
16              ;       SQQR    ALUFUNC,SP,DR                   NEXT UF/TEST CL
17              ;       SZQR    ALUFUNC,SP,DR                   NEXT UF/TEST CL
18
19              ;-------------------------------------------------------
```

```
10015 .MAIN
01
02              ;*************************************************************
03              ;FORMAT 1: ARITHMETRIC-LOGIC INSTRUCTIONS WITH INT. SOURCE
04              ; REGISTER AND INT./EXT. DESTINATION REGISTER
05              ;*************************************************************
06
07
08              ;-------------------------------------------------------------
09              ;NO LOAD
10
11              ;       AQ      ALUFUNC,GRA                             NEXT UF/TEST CL
12              ;       AB      ALUFUNC,GRA,GRB                         NEXT UF/TEST CL
13              ;       ZQ      ALUFUNC                                 NEXT UF/TEST CL
14              ;       ZB      ALUFUNC,GRB                             NEXT UF/TEST CL
15              ;       ZA      ALUFUNC,GRA                             NEXT UF/TEST CL
16              ;       RA      ALUFUNC,SR,GRA                          NEXT UF/TEST CL
17              ;       RQ      ALUFUNC,SR                              NEXT UF/TEST CL
18              ;       RZ      ALUFUNC,SR                              NEXT UF/TEST CL
19
20              ;-------------------------------------------------------------
21
10016 .MAIN
01
02              ;-------------------------------------------------------------
03              ;LOAD GRB
04
05              ;       AQB     ALUFUNC,GRA,GRB                         NEXT UF/TEST CL
06              ;       ABB     ALUFUNC,GRA,GRB                         NEXT UF/TEST CL
07              ;       ZQB     ALUFUNC,GRB                             NEXT UF/TEST CL
08              ;       ZBB     ALUFUNC,GRB                             NEXT UF/TEST CL
09              ;       ZAB     ALUFUNC,GRA,GRB                         NEXT UF/TEST CL
10              ;       RAB     ALUFUNC,SR,GRA,GRB                      NEXT UF/TEST CL
11              ;       RQB     ALUFUNC,SR,GRB                          NEXT UF/TEST CL
12              ;       RZB     ALUFUNC,SR,GRB                          NEXT UF/TEST CL
13
14              ;-------------------------------------------------------------
15
```

```
10017 .MAIN
01              ;-------------------------------------------------------
02              ;LOAD Q
03
04              ;       AQQ     ALUFUNC,GRA                             NEXT UF/TEST CL
05              ;       ABQ     ALUFUNC,GRA,GRB                         NEXT UF/TEST CL
06              ;       ZQQ     ALUFUNC                                 NEXT UF/TEST CL
07              ;       ZBQ     ALUFUNC,GRB                             NEXT UF/TEST CL
08              ;       ZAQ     ALUFUNC,GRA                             NEXT UF/TEST CL
09              ;       RAQ     ALUFUNC,SR,GRA                          NEXT UF/TEST CL
10              ;       RQQ     ALUFUNC,SR                              NEXT UF/TEST CL
11              ;       RZQ     ALUFUNC,SR                              NEXT UF/TEST CL
12
13              ;-------------------------------------------------------
14
10018 .MAIN
01
02              ;-------------------------------------------------------
03              ;LOAD DESTINATION REGISTER
04
05              ;       AQR     ALUFUNC,GRA,DR                          NEXT UF/TEST CL
06              ;       ABR     ALUFUNC,GRA,GRB,DR                      NEXT UF/TEST CL
07              ;       ZQR     ALUFUNC,DR                              NEXT UF/TEST CL
08              ;       ZBR     ALUFUNC,GRB,DR                          NEXT UF/TEST CL
09              ;       ZAR     ALUFUNC,GRA,DR                          NEXT UF/TEST CL
10              ;       RAR     ALUFUNC,SR,GRA,DR                       NEXT UF/TEST CL
11              ;       RQR     ALUFUNC,SR,DR                           NEXT UF/TEST CL
12              ;       RZR     ALUFUNC,SR,DR                           NEXT UF/TEST CL
13
14              ;-------------------------------------------------------
15
```

```
10019 .MAIN
01
02              ;-------------------------------------------------------
03              ;LOAD GRB, DESTINATION REGISTER
04
05              ;       AQBR   ALUFUNC,GRA,GRB,DR                NEXT UF/TEST CL
06              ;       ABBR   ALUFUNC,GRA,GRB,DR                NEXT UF/TEST CL
07              ;       ZQBR   ALUFUNC,GRB,DR                    NEXT UF/TEST CL
08              ;       ZBBR   ALUFUNC,GRB,DR                    NEXT UF/TEST CL
09              ;       ZABR   ALUFUNC,GRA,GRB,DR                NEXT UF/TEST CL
10              ;       RABR   ALUFUNC,SR,GRA,GRB,DR             NEXT UF/TEST CL
11              ;       RQBR   ALUFUNC,SR,GRB,DR                 NEXT UF/TEST CL
12              ;       RZBR   ALUFUNC,SR,GRB,DR                 NEXT UF/TEST CL
13
14              ;-------------------------------------------------------
10020 .MAIN
01
02              ;-------------------------------------------------------
03              ;LOAD Q, DESTINATION REGISTER
04
05              ;       AQQR   ALUFUNC,GRA,DR                    NEXT UF/TEST CL
06              ;       ABQR   ALUFUNC,GRA,GRB,DR                NEXT UF/TEST CL
07              ;       ZQQR   ALUFUNC,DR                        NEXT UF/TEST CL
08              ;       ZBQR   ALUFUNC,GRB,DR                    NEXT UF/TEST CL
09              ;       ZAQR   ALUFUNC,GRA,DR                    NEXT UF/TEST CL
10              ;       RAQR   ALUFUNC,SR,GRA,DR                 NEXT UF/TEST CL
11              ;       RQQR   ALUFUNC,SR,DR                     NEXT UF/TEST CL
12              ;       RZQR   ALUFUNC,SR,DR                     NEXT UF/TEST CL
13
14              ;-------------------------------------------------------
15
10021 .MAIN
01
02              ;-------------------------------------------------------
03              ;EXCEPTION MANIPULATION OF GRA, LOAD GRB
04
05              ;       EAB    ALUFUNC,EX-FUNC,GRA,GRB           NEXT UF/TEST CL
06              ;       EQB    ALUFUNC,EX-FUNC,GRA,GRB           NEXT UF/TEST CL
07              ;       EZB    ALUFUNC,EX-FUNC,GRA,GRB           NEXT UF/TEST CL
08
09              ;-------------------------------------------------------
```

```
10022 .MAIN
01
02              ;************************************************
03              ;FORMAT 2: ARITHMETRIC-LOGIC INSTRUCTIONS WITH
04              ; CONSTANT ADDRESS SOURCE.
05              ;************************************************
06
07
08              ;-------------------------------------------------------
09              ;NO LOAD
10
11              ;       CA      ALUFUNC,CA,GRA                  NEXT UF/TEST CL
12              ;       CQ      ALUFUNC,CA                      NEXT UF/TEST CL
13
14              ;-------------------------------------------------------
15
16
17              ;-------------------------------------------------------
18              ;LOAD GRB
19
20              ;       CAB     ALUFUNC,CA,GRA,GRB              NEXT UF/TEST CL
21              ;       CQB     ALUFUNC,CA,GRB                  NEXT UF/TEST CL
22              ;       CZB     ALUFUNC,CA,GRB                  NEXT UF/TEST CL
23
24              ;-------------------------------------------------------
10023 .MAIN
01
02              ;-------------------------------------------------------
03              ;LOAD Q
04
05              ;       CAQ     ALUFUNC,CA,GRA                  NEXT UF/TEST CL
06              ;       CQQ     ALUFUNC,CA                      NEXT UF/TEST CL
07              ;       CZQ     ALUFUNC,CA                      NEXT UF/TEST CL
08
09              ;-------------------------------------------------------
10
```

```
!0024 .MAIN
01              ;-------------------------------------------------------
02              ;LOAD DESTINATION REGISTER
03
04              ;       CAR     ALUFUNC,CA,GRA,DR                       NEXT UF/TEST CL
05              ;       CQR     ALUFUNC,CA,DR                           NEXT UF/TEST CL
06              ;       CZR     ALUFUNC,CA,DR                           NEXT UF/TEST CL
07
08              ;-------------------------------------------------------
09
10
11              ;-------------------------------------------------------
12              ;LOAD GRB, DESTINATION REGISTER
13
14              ;       CABR    ALUFUNC,CA,GRA,GRB,DR                   NEXT UF/TEST CL
15              ;       CQBR    ALUFUNC,CA,GRB,DR                       NEXT UF/TEST CL
16              ;       CZBR    ALUFUNC,CA,GRB,DR                       NEXT UF/TEST CL
17
18              ;-------------------------------------------------------
19
!0025 .MAIN
01
02              ;*******************************************************
03              ;FORMAT 3: ARITHMETRIC-LOGIC INSTRUCTIONS WITH EXTERNAL
04              ; SOURCE REGISTER.
05              ;*******************************************************
06
07
08              ;-------------------------------------------------------
09              ;NO LOAD
10
11              ;       XA      ALUFUNC,XR,GRA                          NEXT UF/TEST CL
12              ;       XQ      ALUFUNC,XR                              NEXT UF/TEST CL
13              ;       XZ      ALUFUNC,XR                              NEXT UF/TEST CL
14
15              ;-------------------------------------------------------
```

```
10026 .MAIN
01
02              ;-------------------------------------------------------
03              ;LOAD GRB
04
05              ;       XAB     ALUFUNC,XR,GRA,GRB                      NEXT UF/TEST CL
06              ;       XQB     ALUFUNC,XR,GRB                          NEXT UF/TEST CL
07              ;       XZB     ALUFUNC,XR,GRB                          NEXT UF/TEST CL
08              ;       XZAB    XR,ALUFUNC,GRA,GRB                      NEXT UF/TEST CL
09              ;       XABB    XR,ALUFUNC,GRA,GRB                      NEXT UF/TEST CL
10              ;       XXZQ    XR,ALUFUNC                              NEXT UF/TEST CL
11
12              ;-------------------------------------------------------
13
10027 .MAIN
01
02              ;-------------------------------------------------------
03              ;LOAD Q
04
05              ;       XAQ     ALUFUNC,XR,GRA                          NEXT UF/TEST CL
06              ;       XQQ     ALUFUNC,XR                              NEXT UF/TEST CL
07              ;       XZQ     ALUFUNC,XR                              NEXT UF/TEST CL
08
09              ;-------------------------------------------------------
10
11              ;-------------------------------------------------------
12              ;LOAD DESTINATION REGISTER
13
14              ;       XAR     ALUFUNC,XR,GRA,DR                       NEXT UF/TEST CL
15              ;       XQR     ALUFUNC,XR,DR                           NEXT UF/TEST CL
16              ;       XZR     ALUFUNC,XR,DR                           NEXT UF/TEST CL
17              ;       XZAR    XR,ALUFUNC,GRA,DR                       NEXT UF/TEST CL
18
19              ;-------------------------------------------------------
20
```

```
10028 .MAIN
01
02              ;-------------------------------------------------------
03              ;LOAD GRB, DESTINATION REGISTER
04
05              ;       XABR   ALUFUNC,XR,GRA,GRB,DR              NEXT UF/TEST CL
06              ;       XQBR   ALUFUNC,XR,GRB,DR                  NEXT UF/TEST CL
07              ;       XZBR   ALUFUNC,XR,GRB,DR                  NEXT UF/TEST CL
08
09              ;-------------------------------------------------------
10
10029 .MAIN
01              ;-------------------------------------------------------
02              ;LOAD Q, DESTINATION REGISTER
03
04              ;       XAQR   ALUFUNC,XR,GRA,DR            NEXT UF/TEST CL
05              ;       XQQR   ALUFUNC,XR,DR                NEXT UF/TEST CL
06              ;       XZQR   ALUFUNC,XR,DR                NEXT UF/TEST CL
07
08              ;-------------------------------------------------------
09
```

```
1003C .MAIN
01              ;*************************************************************
02              ;FORMAT 4: ARITHMETRIC-LOGIC INSTRUCTIONS WITH HALF-WORD
03              ; MANIPULATION OF GRA REGISTER.
04              ;*************************************************************
05
06
07              ;-------------------------------------------------------------
08              ;LOAD GRB
09
10              ;       HAB     ALUFUNC,HW-FUNC,GRA,GRB             NEXT CL
11              ;       HQB     ALUFUNC,HW-FUNC,GRA,GRB             NEXT CL
12              ;       HZB     ALUFUNC,HW-FUNC,GRA,GRB             NEXT CL
13
14              ;-------------------------------------------------------------
15
16
17              ;*************************
18              ;FORMAT 5: LOAD IMMEDIATE
19              ;*************************
20
21              ;-------------------------------------------------------------
22              ;NO LOAD
23
24              ;       IQ      ALUFUNC,MS-HW,LS-HW                 NEXT CL
25
26              ;-------------------------------------------------------------
27
10031 .MAIN
01              ;-------------------------------------------------------------
02              ;LOAD Q
03
04              ;       IQQ     ALUFUNC,MS-HW,LS-HW                 NEXT CL
05              ;       IZQ     ALUFUNC,MS-HW,LS-HW                 NEXT CL
06
07              ;-------------------------------------------------------------
08
```

```
!0032 .MAIN
01              ;****************************************
02              ;FORMAT 6: CONDITIONAL JUMP INSTRUCTIONS.
03              ;****************************************
04
05
06              ;-------------------------------------------------------
07              ;CONDITIONAL JUMP.
08
09              ;       JMP    T/F,COND,ADDRESS                    NEXT CL
10              ;       JXAQ   ALUFUNC,GRA,T/F,COND                NEXT CL
11              ;       JXZQ   ALUFUNC,T/F,COND                    NEXT CL
12              ;       JXZA   ALUFUNC,GRA,T/F,COND                NEXT CL
13
14              ;-------------------------------------------------------
15
!0033 .MAIN
01              ;-------------------------------------------------------
02              ;CONDITIONAL SUBROUTINE CALLS.
03
04              ;       CALL   T/F,COND,ADDRESS                    NEXT CL
05              ;       CXAQ   ALUFUNC,GRA,T/F,COND                NEXT CL
06              ;       CXZQ   ALUFUNC,T/F,COND                    NEXT CL
07              ;       CXZA   ALUFUNC,GRA,T/F,COND                NEXT CL
08
09              ;-------------------------------------------------------
10
!0034 .MAIN
01              ;-------------------------------------------------------
02              ;CONDITIONAL SUBROUTINE CALLS, LOAD Q
03
04              ;       CXZQQ  ALUFUNC,T/F,COND                    NEXT CL
05              ;       CXAQQ  ALUFUNC,GRA,T/F,COND                NEXT CL
06
07              ;-------------------------------------------------------
08
```

```
10035 .MAIN
01              ;*************************************************
02              ; FORMAT 7: SHIFT, MULTIPLY AND DIVIDE INSTRUCTIONS
03              ;*************************************************
04
05              ;-------------------------------------------------------
06              ;SHIFT LEFT GRB, CONDITION TEST
07
08              ;       SLABB ALUFUNC,GRA,GRB,SI,TST,T/F,COND    NEXT CL
09              ;       SLZBB ALUFUNC,GRB,SI,TST,T/F,COND        NEXT CL
10              ;       SLZAB ALUFUNC,GRA,GRB,SI,TST,T/F,COND    NEXT CL
11
12              ;-------------------------------------------------------
13
14
15              ;-------------------------------------------------------
16              ;SHIFT LEFT GRB CON Q, CONDITION TEST
17
18              ;       DLABB ALUFUNC,GRA,GRB,SI,TST,T/F,COND    NEXT CL
19              ;       DLZBB ALUFUNC,GRB,SI,TST,T/F,COND        NEXT CL
20              ;       DLZAB ALUFUNC,GRA,GRB,SI,TST,T/F,COND    NEXT CL
21
22              ;-------------------------------------------------------
23
```

```
10036 .MAIN
01              ;-------------------------------------------------------
02              ;SHIFT RIGHT GRB, CONDITION TEST
03
04              ;       SRABB ALUFUNC,GRA,GRB,SI,TST,T/F,COND   NEXT CL
05              ;       SRZBB ALUFUNC,GRB,SI,TST,T/F,COND       NEXT CL
06              ;       SRZAB ALUFUNC,GRA,GRB,SI,TST,T/F,COND   NEXT CL
07
08              ;-------------------------------------------------------
09
10
11              ;-------------------------------------------------------
12              ;SHIFT RIGHT GRB CON Q, CONDITION TEST
13
14              ;       DRABB ALUFUNC,GRA,GRB,SI,TST,T/F,COND   NEXT CL
15              ;       DRZBB ALUFUNC,GRB,SI,TST,T/F,COND       NEXT CL
16              ;       DRZAB ALUFUNC,GRA,GRB,SI,TST,T/F,COND   NEXT CL
17
18              ;-------------------------------------------------------
19
```

```
10037 .MAIN
01
02              ;-------------------------------------------------------
03              ;NO SHIFT, LOAD GRB, CONDITION TEST
04
05              ;       NSAQB ALUFUNC,GRA,GRB,TST,T/F,COND       NEXT CL
06              ;       NSABB ALUFUNC,GRA,GRB,TST,T/F,COND       NEXT CL
07              ;       NSZQB ALUFUNC,GRB,TST,T/F,COND           NEXT CL
08              ;       NSZBB ALUFUNC,GRB,TST,T/F,COND           NEXT CL
09              ;       NSZAB ALUFUNC,GRA,GRB,TST,T/F,COND       NEXT CL
10
11              ;-------------------------------------------------------
12
10038 .MAIN
01              ;-------------------------------------------------------
02              ;MULTIPLY
03
04              ;       MULD  ALUFUNC,GRA,GRB,SI,TST,T/F,COND    NEXT CL; MULTIPLY,
05                                                                   ; DOUBLE SHIFT
06              ;       MULS  ALUFUNC,GRA,GRB,SI,TST,T/F,COND    NEXT CL; MULTIPLY,
07                                                                   ; SINGLE SHIFT
08              ;       MULN  ALUFUNC,GRA,GRB,TST,T/F,COND       NEXT CL; MULTIPLY,
09                                                                   ; NO SHIFT
10              ;       MULQ  ALUFUNC,GRA,TST,T/F,COND           NEXT CL; MULTIPLY,
11                                                                   ; WITH Q-REGISTER
12
13              ;-------------------------------------------------------
14
10039 .MAIN
01              ;-------------------------------------------------------
02              ;DIVIDE
03
04              ;       DIVD  ALUFUNC,GRA,GRB,SI,TST,T/F,COND    NEXT CL
05              ;       DIVS  ALUFUNC,GRA,GRB,SI,TST,T/F,COND    NEXT CL
06              ;       DIVN  ALUFUNC,GRA,GRB,TST,T/F,COND       NEXT CL
07
08              ;-------------------------------------------------------
09
 004C .MAIN
0000 SOURCE LINES IN ERROR
```

```
 0001 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;FILE: T1, EDITED: 850814/LBJ
03
04                    00001 .NOMA 1
05                    00000 .LOC 0
06                          ;MICRO INTERRUPT TABLE
07                          ;**********************
08
09                          ;THE FIRST 7 LOCATIONS CONTAINS A JUMP TABLE FOR
10                          ;INTERRUPT AND ERROR ACTIONS
11
12 00000 4631000000004000          JMP   F,F,POWUP                          ;0: POWER UP
13 00001 4631000000000023          JMP   F,F,INTIN                          ;1: CPU INTERRUPT
14 00002 0631000000001661          JMP   F,F,INT00                          ;2: LIMIT VIOLATION
15 00003 4631000000001415          JMP   F,F,INTBB                          ;3: I/O ERROR
16 00004 4631000000001671          JMP   F,F,INT1                           ;4: INTEGER MASK & OVERFLOW
17 00005 4631000000001671          JMP   F,F,INT1                           ;5: INTEGER MASK & SHIFT OVERFLOW
18                                                                          ;6: EXTERNAL INTERRUPT
19 00006 0605230000000022          ZOR   OR,ILEVR                           ; ILEVR:= 0
20                                                                          ; DELAY 675 NSEC BEFORE ILEV IS READ
21 00007 0715730000004100          XZ    OR,IOSTA              CONT 0 2     ; TERMINATE A POSSIBLE READ
22 00010 4705230000000000          ZO    OR                    CONT 0 2     ;
23 00011 4724730000007760          IZO   OR,0,377              CONT 2       ;
24 00012 0617640032000320          XOBR  AND,ILEV,WRK0,INTRG                ; WRK0:=INTRG:= ILEV(16:23)
25 00013 4637431525000060          SLZAB OR,WRK0,CAUSE,Z,NL,F,F             ; CAUSE:= WRK0 SHIFT 1
26 00014 4651523500003600          CA    SUB0,,8,WRK0                       ; IF LEVEL=8 THEN
27 00015 4631000000421434          JMP   T,NZ,ICALL                         ; BEGIN
28 00016 4605431500000017          ZAR   OR,WRK0,RTCR                       ;  <* INTERRUPT FROM OTHER PU *>
29 00017 0731000000000040          JMP   F,F,DCLOK             CONT 2       ;  <* SAVE RTC IN PUTAB+PUINDEX *>
```

```
10002 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          AUXIN:
03                          ;IDLE:                                                  ;IDLE:
04                          ;MUST HAVE LOCATION 8'20
05 00020 0653720031000706          CZBR  SUB,.32,CNTR,CNTRO                        ; CNTR:=CNTRO:= 8'37
06 00021 4631000002612000          CALL  T,EXINT,CHCLK                             ; IF EXTERNAL INTERRUPT THEN
07                                                                                 ;   CHECK CLOCK INTERRUPT;
08 00022 4631000000200021          JMP   F,CPUIN,.-3                               ; WAIT FOR CPU INTERRUPT
09
10                          ;INTIN:                                                 ;CPU INTERRUPTS: C. LOC 23
11 00023 0631000000224001          JMP   F,NTCPI,TCPIN                             ; TCP INPUT
12 00024 0631000000230020          JMP   F,NAUXI,AUXIN                             ; AUX INTERRUPT
13 00025 0631000000244020          JMP   F,NCAME,CAMER                             ; CAM ERROR
14 00026 4631000000254002          JMP   F,NOCPA,OCPAL                             ; OCP AOTOLOAD
15 00027 0631000000264003          JMP   F,NREMA,REMAL                             ; REMOTE AUTOLOAD
16 00030 0631000000004004          JMP   F,F,SINGL                                 ; SINGLE INSTRUCTION INTERRUPT
17 00031 0631000000001632          JMP   F,F,.SCAP                                 ; GOTO ESCAPE C. LOC 8'31
18 00032 4631000000001665          JMP   F,F,.NTO                                  ; GOTO INTO
19 00033 0631000000001424          JMP   F,F,.NTB1                                 ; GOTO INTB1
20 00034 0631000000001771          JMP   F,F,.YFL1                                 ; GOTO SYFL1
21 00035 0631000000001667          JMP   F,F,.NTF                                  ; GOTO INTF
22 00036 4631000000001401          JMP   F,F,.FINT                                 ; GOTO PFINT
23 00037 4631000000001533          JMP   F,F,LOKM                                  ; GOTO LOCKMONITOR
24
25                          DCLOK:                                                  ;
26 00040 0615730000000212          XZR   OR,RTC,DATO                               ;  <* DELAY 200 NSEC *>
27 00041 0600730000001100          SZQ   OR,PUTAB                                  ;
28 00042 4640600000001000          SQQ   ADD,PUIX                                  ;
29 00043 0651600000231311          CQR   ADD,.10,LAD          CONT WRT             ;  LAD:= PUTAB+PUIX+10, WRT;
30 00044 0655730000510100          XZ    OR,CIOST             CONT TSTIO 1         ; TESTIO;
31 00045 0611541440001500          CA    AND,.4,CNTR                               ;  IF CNTR(21)<>0 THEN <* NOT RUN*>
32 00046 0631000000422001          JMP   T,NZ,DISPA                                ;     GOTO DISPA;
33 00047 4605730000241011          RZR   OR,ICS,LAD           CONT READJ           ;  LAD:= ICS, READ INSTRUCTION;
34 00050 3555730000510100          XZ    OR,CIOST             NEXT TSTIO 3         ; TESTIO, NEXT INSTRUCTION
```

```
10003 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;INSTRUCTION EXECUTION PROCEDURES
03                          ;*********************************
04
05                          ;ADDRESS HANDLING
06                          ;*****************
07
08                          ;MODIFY NEXT ADDRESS, AM, OPCODE = 11
09
10                          AME:                                                ;AME:
11 00051 4624730000022020           IZQ   OR,0,1101                           ;
12 00052 0631000002000031           CALL  F,F,ESCAP                           ; CALL ESCAPE(1101)
13                          AM:                                                 ;AM:
14 00053 4613531224003205           CABR  OR,.182,STAT,STAT,CPUST             ; AFTER AM:= 1
15 00054 3445230000600000           ZQ    OR                        NEXT TSTIN 1 ;TEST INT, NEXT INSTRUCTION
```

```
10004 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;LOAD ADDRESS, AL, OPCODE = 13
03
04                          ALE:                                                ;ALE:
05 CC055 0624730000026020           IZQ    OR,0,1301                        ;
06 CC056 0631000002000031           CALL   F,F,ESCAP                        ; CALL ESCAPE(1301)
07
08                          AL:                                                 ;AL:
09 CC057 7407431300600000           ZAB    OR,SB,W                  NEXT TSTIN ; W:=SB, TEST INTERRUPT
10                                                                          ; NEXT INSTRUCTION
11
12                          ;LOAD ADDRESS COMPLEMENTED, AC, OPCODE = 41
13
14                          ACE:                                                ;ACE:
15 CC060 4624730000102020           IZQ    OR,0,4101                        ;
16 CC061 0631000002000031           CALL   F,F,ESCAP                        ; CALL ESCAPE(4101)
17                          AC:                                                 ;AC:
18 CC062 0607423300000000           ZAB    SUB0,SB,W                        ; W:= -SB
19 CC063 7406731224644000           EZB    OR,STOC,STAT,STAT        NEXT TOVIN ; EX(22:23):=OVERFLOW,CARRY
20                                                                          ; TEST OVERFLOW, TEST INT
21                                                                          ; NEXT INSTRUCTION
```

```
1CCC5 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;WORD INSTRUCTIONS WITH MEMORY REFERENCE
03                          ;**************************************
04
05                          ;LOAD REGISTER, RL, OPCODE = 24
06
07                          RLE:                                                ;RLE:
08 CC064 4624730000C50040           IZQ    OR,0,2402                            ;
09 CC065 0631000002CC0031           CALL   F,F,ESCAP                            ; CALL ESCAPE(2402)
10                          RL:                                                 ;RL:
11 CC066 0631000000130070           JMP    F,NWADR,RLREG                        ; IF W-ADDR THEN GOTO RL-REGISTER
12 CC067 7457730000714000           XZB    OR,DATI,W              NEXT TIOIN 1  ; W:=DATAIN, TEST IO, TEST INT
13                                                                              ; GOTO NEXT INSTRUCTION
14 CC070 3457430140600000 RLREG:    XZAB   CDATI,OR,LAX,W         NEXT TSTIN 1  ; W:=LAX, TEST INT
15                                                                              ; GOTO NEXT INSTRUCTION
16
17                          ;LOGICAL AND, LA, OPCODE = 4
18
19                          LAE:                                                ;LAE:
20 CC071 0624730000C10040           IZQ    OR,0,0402                            ;
21 CC072 0631000002CC0031           CALL   F,F,ESCAP                            ; CALL ESCAPE(0402)
22                          LA:                                                 ;LA:
23 CC073 0631000000130075           JMP    F,NWADR,LAREG                        ; IF W-ADDR THEN GOTO LA-REG
24 CC074 7457540000714000           XAB    AND,DATI,W,W           NEXT TIOIN 1  ; W:= W AND DATAIN, TEST INT, TEST IO
25                                                                              ; GOTO NEXT INSTRUCTION
26 CC075 7457140140600000 LAREG:    XABB   CDATI,AND,LAX,W        NEXT TSTIN 1  ; W:=LAX AND W, TEST INT
27                                                                              ; GOTO NEXT INSTRUCTION
28
29                          ;LOGICAL OR, LO, OPCODE = 5
30
31                          LOE:                                                ;LOE:
32 CC076 4624730000012040           IZQ    OR,0,0502                            ;
33 CC077 0631000002CC0031           CALL   F,F,ESCAP                            ; CALL ESCAPE(0502)
34                          LO:                                                 ;LO:
35 CC100 4631000000130102           JMP    F,NWADR,LOREG                        ; IF W-ADDR THEN GOTO LO-REG
36 CC101 3457530000714000           XAB    OR,DATI,W,W            NEXT TIOIN 1  ; W:=W OR DATAIN, TEST IO, TEST INT
37                                                                              ; GOTO NEXT INSTRUCTION
38 CC102 3457130140600000 LOREG:    XABB   CDATI,OR,LAX,W         NEXT TSTIN 1  ; W:= W OR LAX, TEST INT
39                                                                              ; GOTO NEXT INSTRUCTION
```

```
10CC6 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;LOGICAL EXCLUSIVE OR, LX, OPCODE = 6
03
04                          LXE:                                                ;LXE:
05 CC103 4624730000014040           IZQ    OR,0,0602                        ;
06 CC104 0631000002CC0031           CALL   F,F,ESCAP                        ; CALL ESCAPE(C602)
07                          LX:                                                 ;LX:
08 CC105 4631000000130107           JMP    F,NWADR,LXREG                    ; IF W-ADDR THEN GOTO LX-REG
09 CC106 3457560000714000           XAB    EXOR,DATI,W,W          NEXT TIOIN 1 ; W:=W EXOR DATAIN, TEST INT, TEST IO
10                                                                          ; GOTO NEXT INSTRUCTION
11 CC107 345716014060C000 LXREG:    XABB   CDATI,EXOR,LAX,W       NEXT TSTIN 1 ; W:= W EXOR LAX, TEST INT
12                                                                          ; GOTO NEXT INSTRUCTION
13
14                          ;ADD INTEGER WORD, WA, OPCODE = 7
15
16                          WAE:                                                ;WAE:
17 CC110 0624730000C16040           IZQ    OR,0,702                         ;
18 CC111 0631000002CC0031           CALL   F,F,ESCAP                        ; CALL ESCAPE(C702)
19                          WA:                                                 ;WA:
20 CC112 4631000000130115           JMP    F,NWADR,WAREG                    ; IF W-ADDR THEN GOTO WA-REG
21 CC113 0717500000514000           XAB    ADD,DATI,W,W           CONT TSTIO 2 ; W:=W + DATAIN, TEST IO
22 CC114 7406731224644000           EZB    OR,STOC,STAT,STAT      NEXT TOVIN   ; STATUS:=STATUS(0:21) CON OVERFLOW,CARRY
23                                                                          ; TEST OVERFLOW, TEST INT
24                                                                          ; GOTO NEXT INSTRUCTION
25 CC115 465710014000C000 WAREG:    XABB   CDATI,ADD,LAX,W        CONT 0 1     ; W:=W + LAX
26 CC116 7406731224644000           EZB    OR,STOC,STAT,STAT      NEXT TOVIN   ; STATUS:=STATUS(0:21) CON OVERFLOW, CARRY
27                                                                          ; TEST OVERFLOW, TEST INT
28                                                                          ; GOTO NEXT INSTRUCTION
29
30                          ;SUBTRACT INTEGER WORD, WS, OPCODE = 10
31
32                          WSE:                                                ;WSE:
33 CC117 0624730000020040           IZQ    OR,0,1002                        ;
34 CC120 0631000002CC0031           CALL   F,F,ESCAP                        ; CALL ESCAPE(1002)
35                          WS:                                                 ;WS:
36 CC121 0631000000130124           JMP    F,NWADR,WSREG                    ; IF W-ADDR THEN GOTO WS-REG
37 CC122 0717512000514000           XAB    SUNO,DATI,W,W          CONT TSTIO 2 ; W:=W - DATAIN, TEST IO
38 CC123 7406731224644000           EZB    OR,STOC,STAT,STAT      NEXT TOVIN   ; STATUS:= STATUS(0:21) CON OVERFLOW,CARRY
39                                                                          ; TEST OVERFLOW, TEST INT
40                                                                          ; GOTO NEXT INSTRUCTION
```

```
10007 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 CC124 4657112140000000 WSREG: XABB  CDATI,SUNO,LAX,W      CONT 0 1     ; W:=W - LAX
03 CC125 7406731224644000        EZB  OR,STOC,STAT,STAT     NEXT TOVIN   ; STATUS:=STATUS(0:21) CON OVERFLOW,CARRY
04                                                                       ; TEST OVERFLOW, TEST INT
05                                                                       ; GOTO NEXT INSTRUCTION
06
07                       ;MULTIPLY INTEGER WORD, WM, OPCODE = 12
08
09                       WME:                                            ;WME:
10 CC126 4624730000024040        IZQ  OR,0,1202                          ;
11 CC127 0631000002000031        CALL F,F,ESCAP                          ; CALL ESCAPE(1202)
12                       WM:                                             ;WM:
13 CC130 0631000000130141        JMP  F,NWADR,WMREG                      ; IF W=ADDR THEN GOTO WM-REG
14 CC131 0657730032514000        XZB  OR,DATI,WRK0          CONT TSTIO 1 ; WRK0:= DATAIN , TEST IO
15 CC132 4604430000C00000 WM1:   ZAQ  OR,W                              ; Q:= W
16 CC133 4634340001000040        DRZBB AND,WPRE,Z,MC,F,F                 ; ADC:= IF Q(23)=1 THEN 0 ELSE 1
17                                                                       ; WPRE:= 0
18 CC134 0651712000000604        CZR  SUNO,.21,LCD                       ; FOR LCD:= -22 STEP 1 UNTIL 0 DO
19 CC135 4274101501113240        MULD ADD,WRK0,WPRE,SGN,MC,F,LCN PUSH    ;
20 CC136 2634101501113240        MULD ADD,WRK0,WPRE,SGN,MC,F,LCN LRTN 0 0 ; BEGIN
21                                                                       ;   WPRE:= IF ADC=0 THEN WPRE+WRK0
22                                                                       ;   ELSE                  WPRE-0
23                                                                       ;   ADC:=IF Q(23)=1 THEN 0 ELSE 1
24                                                                       ;   WPRE,Q:= WPRE(0) CON WPRE,Q SHIFT (-1
25                                                                       ; END
26 CC137 4674113501003260        MULD SUNO,WRK0,WPRE,SGN,NL,F,F          ; WPRE:=IF ADC=0 THEN WPRE-WRK0
27                                                                       ;                ELSE WPRE-0
28                                                                       ; WPRE,Q:= WPRE(0) CON WPRE,Q SHIFT (-1)
29 CC140 3407230000600000        ZQB  OR,W                  NEXT TSTIN   ; W:= Q , TEST INT
30                                                                       ; GOTO NEXT INSTR
31
32                       WMREG:                                          ;WM-REG:
33 CC141 0657430172000000        XZAB CDATI,OR,LAX,WRK0     CONT 0 1     ; WRK0:= LAX
34 CC142 0631000000000132        JMP  F,F,WM1                            ; GOTO WM1
35
```

```
1CCC8 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;MULTIPLY INTEGER WORD, WM, WHEN THE FPU801 IS PRESENT
03
04                          WMFPE:                                                  ;WMFPE:
05 CC143 4624730000024040           IZQ    OR,0,1202                            ;
06 CC144 0631000002000031           CALL   F,F,ESCAP                            ; CALL ESCAPE(1202)
07                          WMFPU:                                                  ;WMFPU:
08 CC145 4605230000040071           ZQR    OR,FPOP1              CONT WMOP      ; FPUOP1:= Q   , WMOP
09 CC146 4605430000000070           ZAR    OR,W,FPOPO                           ; FPUOPU:= W   , START FPU
10 CC147 0631000000130156           JMP    F,NWADR,WMFPR                        ; IF W-ADDR THEN GOTO WMFPU-REG
11 CC150 4717730032004000           XZB    OR,DATI,WRKO          CONT O 2       ; WRKO:= DATAIN
12 CC151 0605431500000070           ZAR    OR,WRKO,FPOPU                        ; FPUOPU:= WRKO  , START FPU
13 CC152 4615730000514000           XZ     OR,DATI               CONT TSTIO     ; TEST IO
14 CC153 0615730000007000           XZ     OR,FPURU                             ; WAIT FOR FPURO TO BE BUSY
15 CC154 4657730001007000           XZB    OR,FPURU,WPRE         CONT O 1       ; WPRE:= FPURO
16 CC155 3457730000607100           XZB    OR,FPUR1,W            NEXT TSTIN 1   ; W:= FPUR1   , TEST INT
17                                                                               ; GOTO NEXT INSTR
18                          WMFPR:                                                  ;WMFPU-REG:
19 CC156 0655230000000000           XXZQ   CDATI,OR              CONT O 1       ; NO-OP
20 CC157 4605430140000070           ZAR    OR,LAX,FPOPO                         ; FPUOPU:= LAX   , START FPU
21 CC160 0615730000007000           XZ     OR,FPURO                             ; WAIT FOR FPURO TO BE BUSY
22 CC161 4657730001007000           XZB    OR,FPURU,WPRE         CONT O 1       ; WPRE:= FPURO
23 CC162 3457730000607100           XZB    OR,FPUR1,W            NEXT TSTIN 1   ; W:= FPUR1   , TEST INT
24                                                                               ; GOTO NEXT INSTR
25
26                          ; DIVIDE INTEGER WORD, WD, OPCODE = 30
27
28                          WDE:                                                    ;WDE:
29 CC163 4624730000060040           IZQ    OR,U,3002                            ;
30 CC164 0631000002000031           CALL   F,F,ESCAP                            ; CALL ESCAPE(3002)
31                          WD:                                                     ;WD:
32 CC165 4604430000000000           ZAQ    OR,W                                 ; Q:= W
33 CC166 0607430076000000           ZAB    OR,WPRE,WRK4                         ; WRK4:= WPRE
34 CC167 4631000000130223           JMP    F,NWADR,WDREG                        ; IF W-ADDR THEN GOTO WD-REG
35 CC170 0657730032514000           XZB    OR,DATI,WRKO          CONT TSTIO 1   ; WRKU:= DATAIN , TESTIO
```

```
10009 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 CC171 0637330032C04000 WD1:    NSZBB OR,WRK0,DVS,F,F                          ; DVS:= WRK0(0) , DIVISOR SIGN
03 CC172 4631000000020220        JMP   F,NZ,WDOFL                               ; IF WRK0=0 THEN GOTO WD-OVERFLOW
04 CC173 4636330001CC0020        DLZBB OR,WPRE,Z,DC,F,F                         ; ADC:=IF WPRE(0)=DVS THEN 1 ELSE 0
05                                                                              ; WPRE,Q:= WPRE,Q SHIFT 1
06 CC174 0676107501C02120        DIVD  ADDX,WRK0,WPRE,ADC,DC,F,F                ; WPRE:=IF ADC=1 THEN WPRE-WRK0
07                                                                              ;                ELSE WPRE+WRK0
08                                                                              ; ADC:=IF WPRE(0)=DVS THEN 1 ELSE 0
09                                                                              ; WPRE,Q:= WPRE,Q SHIFT 1
10 CC175 0631000000500220        JMP   T,DIVOF,WDOFL                            ; IF CARRY<>LNK THEN GOTO WD-OVERFLOW
11 CC176 0651511400000604        CAR   SUN,.21,.2,LCD                           ; FOR LCD:=-21 STEP 1 UNTIL 0 DO
12 CC177 4276107501112120        DIVD  ADDX,WRK0,WPRE,ADC,DC,F,LCN PUSH         ;
13 CC200 2636107501112120        DIVD  ADDX,WRK0,WPRE,ADC,DC,F,LCN LRTN 0 0     ; BEGIN
14                                                                              ;   WPRE:=IF ADC=1 THEN WPRE-WRK0
15                                                                              ;                  ELSE WPRE+WRK0
16                                                                              ;   ADC:=IF WPRE(0)=DVS THEN 1 ELSE 0
17                                                                              ;   WPRE,Q:= WPRE,Q SHIFT 1
18                                                                              ; END
19 CC201 0636330035002060        DLZBB OR,WRK3,ADC,NL,F,F                       ; Q:= Q(1:23) CON ADC
20 CC202 4677107501004120        DIVN  ADDX,WRK0,WPRE,DC,F,F                    ; WPRE:=ID ADC=1 THEN WPRE-WRK0
21                                                                              ;                ELSE WPRE+WRK0
22                                                                              ; ADC:=IF WPRE(0)=DVS THEN 1 ELSE 0
23 CC203 0636330035002060        DLZBB OR,WRK3,ADC,NL,F,F                       ; Q:= Q(1:23) CON ADC
24 CC204 0677101501C04260        MULN  ADD,WRK0,WPRE,NL,F,F                     ; WPRE:=IF ADC=0 THEN WPRE+WRK0
25                                                                              ;                ELSE WPRE
26 CC205 0605122072CC0000        AB    SUB0,WPRE,WRK0                           ; IF REMAINDER=DIVISOR
27 CC206 4631000000C20213        JMP   F,NZ,WDCOR                               ;   GOTO WDCOR
28 CC207 4605430040CC0000        ZA    OR,WPRE                                  ; IF REMAINDER=0 THEN
29 CC210 4631000000020216        JMP   F,NZ,WDEND                               ;   GOTO WDEND
30 CC211 4605161701CC0000        AB    EXOR,WRK4,WPRE                           ; IF SIGN(DIVIDEND)=SIGN(REMAINDER) THEN
31 CC212 4631000000010216        JMP   F,NEG,WDEND                              ;   GOTO WDEND
32                       WDCOR:                                                 ;WDCOR:
33 CC213 4607113501CC0000        ABB   SUN0,WRK0,WPRE                           ; REMAINDER:=REMAINDER-DIVISOR
34 CC214 0604202000CC0000        ZQQ   ADD0                                     ; QUOTIENT:= QUOTIENT + 1
35 CC215 4631000000430220        JMP   T,OVFL,WDOFL                             ; IF OVERFLOW THEN GOTO WD-OVERFLOW
36                       WDEND:                                                 ;WDEND:
37 CC216 4607230000000000        ZQB   OR,W                                     ; W:= Q
38 CC217 7406731224604200        EZB   OR,STZZ,STAT,STAT      NEXT TSTIN        ; EX:= 0 , TEST INT
39                                                                              ; GOTO NEXT INSTR
```

```
10C1C CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          WDOFL:                                                  ;WD-OVERFLOW:
03 CC220 0607431701CC0000           ZAB   OR,WRK4,WPRE                             ; WPRE:= WRK4  C. RESTORE DIVIDEND
04 CC221 0664713CC0CC0000           IZQ   SUN0,4000,0                             ; SET OVERFLOW
05 CC222 7406731224644000           EZB   OR,STOC,STAT,STAT     NEXT TOVIN        ; EX(22:23):= 1,U  , TEST OVERFLOW
06                                                                                ; TEST INT, GOTO NEXT INSTR
07
08                          WDREG:                                                  ;WD-REG:
09 CC223 0657430172CC0000           XZAB  CDATI,OR,LAX,WRKU     CONT U 1          ; WRKU:= LAX
10 CC224 4631000000CC0171           JMP   F,F,WD1                                 ; GOTO WD1
```

```
10011 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;HALF-WORD INSTRUCTIONS WITH MEMORY REFERENCE
03                          ;*******************************************
04
05                          ;LOAD HALF REGISTER, HL, OPCODE = 3
06
07                          HLE:                                                ;HLE:
08 00225 4624730000006040           IZQ   OR,0,0302                             ;
09 00226 0631000002C00031           CALL  F,F,ESCAP                             ; CALL ESCAPE(0302)
10                          HL:                                                 ;HL:
11 00227 0610550000C0300            CAQ   CAND,B12,W                            ; Q:= W(0:11) CON 12 EXT 0
12 00230 4631000000130234           JMP   F,NWADR,HLREG                         ; IF W-ADDR THEN GOTO HL-REG
13 00231 0657730032514000           XZB   OR,DATI,WRK0          CONT TSTIO 1    ; WRK0:=DATAIN, TEST IO
14 00232 4662631500000000           HQB   OR,HLOAD,WRK0,W                       ; W:= IF ODD THEN
15 00233 7405230000600000           ZQ    OR                    NEXT TSTIN      ;    Q OR 12 EXT 0 CON WRK0(12:23)
16                                                                              ; ELSE
17                                                                              ;   Q OR 12 EXT 0 CON WRK0(0:11)
18                                                                              ; TEST INT, GOTO NEXT INSTR.
19                          HLREG:                                              ;HLREG:
20 00234 0662630140C00000           HQB   OR,HLOAD,LAX,W                        ; W:= IF ODD THEN
21 00235 7455230000600000           XXZQ  CDATI,OR              NEXT TSTIN 1    ;   Q OR 12 EXT 0 CON LAX(12:23)
22                                                                              ; ELSE
23                                                                              ;   Q OR 12 EXT 0 CON LAX(0:11)
24
25                          ;LOAD ZERO EXTENDED HALF-WORD, ZL, OPCODE = 23
26
27                          ZLE:                                                ;ZLE:
28 00236 0624730000046040           IZQ   OR,0,2302                             ;
29 00237 0631000002C00031           CALL  F,F,ESCAP                             ; CALL ESCAPE(2302)
30                          ZL:                                                 ;ZL:
31 00240 0631000000130244           JMP   F,NWADR,ZLREG                         ; IF W-ADDR THEN GOTO ZL-REG
32 00241 0657730032514000           XZB   OR,DATI,WRK0          CONT TSTIO 1    ; WRK0:=DATAIN, TEST IO
33 00242 0662731500C00000           HZB   OR,HLOAD,WRK0,W                       ; W:= IF ODD THEN 12 EXT 0 CON WRK0(12:23
34 00243 3445230000600000           ZQ    OR                    NEXT TSTIN 1    ; ELSE 12 EXT 0 CON WRK0(0:11)
35                                                                              ; TEST INT, GOTO NEXT INSTR
36                          ZLREG:                                              ;ZLREG:
37 00244 4662730140C00000           HZB   OR,HLOAD,LAX,W                        ; W:= IF ODD THEN 12 EXT 0 CON LAX(12:23
38 00245 7455230000600000           XXZQ  CDATI,OR              NEXT TSTIN 1    ; ELSE 12 EXT 0 CON LAX(0:11)
39                                                                              ; TEST INT, GOTO NEXT INSTR
```

```
10012 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;LOAD EXTENDED HALF-WORD, EL, OPCODE = 2
03
04                          ELE:                                                ;ELE:
05 CC246 0624730000004040           IZQ    OR,0,0202                        ;
06 CC247 0631000002CC0031           CALL   F,F,ESCAP                        ; CALL ESCAPE(0202)
07                          EL:                                                 ;EL:
08 CC250 4631000000130254           JMP    F,NWADR,ELREG                    ; IF W-ADDR THEN GOTO EL-REG
09 CC251 0657730032514000           XZB    OR,DATI,WRK0          CONT TSTIO 1 ; WRK0:=DATAIN, TEST IO
10 CC252 4662731500C00100           HZB    OR,EXLOA,WRK0,W                  ; W:= IF ODD THEN
11 CC253 7405230006C00000           ZQ     OR                    NEXT TSTIN   ;   12 EXT WRK0(12) CON WRK0(12:23)
12                                                                           ; ELSE
13                                                                           ;   12 EXT WRK0(0) CON WRK0(0:11)
14                                                                           ; TEST INT, GOTO NEXT INSTR
15                          ELREG:                                              ;ELREG:
16 CC254 0662730140000100           HZB    OR,EXLOA,LAX,W                   ; W:= IF ODD THEN
17 CC255 7455230006C00000           XXZQ   CDATI,OR              NEXT TSTIN 1 ;   12 EXT LAX(12) CON LAX(12:23)
18                                                                           ; ELSE
19                                                                           ;   12 EXT LAX(0) CON LAX(0:11)
20                                                                           ; TEST INT, GOTO NEXT INSTR
21
22                          ;ADD INTEGER HALF-WORD, EA, OPCODE = 22
23
24                          EAE:                                                ;EAE:
25 CC256 4624730000C44040           IZQ    OR,0,2202                        ;
26 CC257 0631000002CC0031           CALL   F,F,ESCAP                        ; CALL ESCAPE(2202)
27                          EA:                                                 ;EA:
28 CC260 4604430000000000           ZAQ    OR,W                             ; Q:=W
29 CC261 0631000000130265           JMP    F,NWADR,EAREG                    ; IF W-ADDR THEN GOTO EA-REG
30 CC262 0657730032514000           XZB    OR,DATI,WRK0          CONT TSTIO 1 ; WRK0:=DATAIN, TEST IO
31 CC263 0722601500000100           HGB    ADD,EXLOA,WRK0,W                 ; W:= IF ODD THEN
32                                                                           ;   Q+ 12 EXT WRK0(12) CON WRK0(12:23)
33                                                                           ; ELSE
34                                                                           ;   Q+ 12 EXT WRK0(0) CON WRK0(0:11)
35 CC264 7406731224644000           EZB    OR,STOC,STAT,STAT     NEXT TOVIN   ; STAT:=STAT(0:21) +OFL+CARRY
36                                                                           ; TEST OFL, TEST INT, GOTO NEXT INSTR
```

```
10013 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          EAREG:                                                  ;EAREG:
03 CC265 4722600140CC0100           HQB   ADD,EXLOA,LAX,W                         ; W:= IF ODD THEN
04 CC266 4606731224444000           EZB   OR,STOC,STAT,STAT      CONT TSTOV     ;   Q + 12 EXT LAX(12) CON LAX(12:23)
05                                                                                ; ELSE
06                                                                                ;   Q + 12 EXT LAX(0) CON LAX(0:11)
07 CC267 3455730000600100           XZ    OR,CIOST               NEXT TSTIN 1 ;STAT:= STAT(0:21)+ OFL+CARRY
08                                                                                ; TEST OFL, TEST INT, GOTO NEXT INSTR
09
10                          ;SUBTRACT INTEGER HALF-WORD, ES, OPCODE = 21
11
12                          ESE:                                                    ;ESE:
13 CC270 4624730060042040           IZQ   OR,0,2102                               ;
14 CC271 0631000002CC0031           CALL  F,F,ESCAP                               ; CALL ESCAPE(2102)
15                          ES:                                                     ;ES:
16 CC272 4604430006CC0000           ZAQ   OR,W                                    ; Q:=W
17 CC273 0631000000130277           JMP   F,NWADR,ESREG                           ; IF W=ADDR THEN GOTO ES-REG
18 CC274 0657730032514000           XZB   OR,DATI,WRK0           CONT TSTIO 1 ; WRK0:=DATAIN, TEST IO
19 CC275 0722613500CC0100           HQB   SUNO,EXLOA,WRK0,W                       ; W:= IF ODD THEN
20                                                                                ;   Q - 12 EXT WRK0(12) CON WRK0(12:23)
21                                                                                ; ELSE
22                                                                                ;   Q - 12 EXT WRK0(0) CON WRK0(0:11)
23 CC276 7406731224644000           EZB   OR,STOC,STAT,STAT      NEXT TOVIN     ; STAT:=STAT(0:21) + OFL + CARRY
24                                                                                ; TEST OFL, TEST INT, GOTO NEXT INSTR
25                          ESREG:                                                  ;ESREG:
26 CC277 4722612140CC0100           HQB   SUNO,EXLOA,LAX,W                        ; W:= IF ODD THEN
27 CC300 4606731224444000           EZB   OR,STOC,STAT,STAT      CONT TSTOV     ;   Q - 12 EXT LAX(12) CON LAX(12:23)
28                                                                                ; ELSE
29 CC301 3455730000600100           XZ    OR,CIOST               NEXT TSTIN 1 ;  Q - 12 EXT LAX(0) CON LAX(0:11)
30                                                                                ; STAT:= STAT(0:21) + OFL + CARRY
31                                                                                ; TEST OFL, TEST INT, GOTO NEXT INSTR
```

```
!0014 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;LOAD EXCEPTION REGISTER, XL, OPCODE = 20
03
04                          XLE:                                                ;XLE:
05 C0302 0624730000040040           IZQ    OR,0,2002                        ;
06 C0303 0631000002C00031           CALL   F,F,ESCAP                        ; CALL ESCAPE(2002)
07                          XL:                                                 ;XL:
08 C0304 4613551224003400           CAB    CAND,,7,STAT,STAT                ; STAT:= STAT(0:21) CON 3EXTC
09 C0305 4631000000130312           JMP    F,NWADR,XLREG                    ; IF W-ADDR THEN GOTO XL-REG
10 C0306 0657730032514000           XZB    OR,DATI,WRK0          CONT TSTIO 1 ; WRK0:= DATAIN, TEST IO
11 C0307 4662731532000000 XL1:      HZB    OR,HLOAD,WRK0,WRK0               ; WRK0:= IF ODD THEN
12                                                                          ;   12 EXT 0 CON WRK0(12:23)
13                                                                          ; ELSE
14                                                                          ;   12 EXT 0 CON WRK0(0:11)
15 C0310 0613541532003400           CAB    AND,,7,WRK0,WRK0                 ; WRK0:= 21 EXT 0 CON WRK0(21:23)
16 C0311 7407131524600000           ABB    OR,WRK0,STAT          NEXT TSTIN ; STAT:= STAT OR WRK0
17                                                                          ; TEST INT, GOTO NEXT INSTR
18                          XLREG:                                              ;XLREG:
19 C0312 0657430172000000           XZAB   CDATI,OR,LAX,WRK0     CONT 0 1   ; WRK0:=LAX
20 C0313 4631000000000307           JMP    F,F,XL1                          ; GOTO XL1
```

```
!0015 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;STORE INSTRUCTIONS
03                          ;*******************
04
05                          ;STORE REGISTER, RS, OPCODE = 27
06
07                          RSE:                                                ;RSE:
08 CC314 4624730000056100           IZQ    OR,U,2704                            ;
09 CC315 0631000002CC0031           CALL   F,F,ESCAP                            ; CALL ESCAPE(2704)
10                          RS:                                                 ;RS:
11 CC316 0605430000CC0012           ZAR    OR,W,DATO                            ; DATAOUT:=W
12 CC317 4613551332223711           CABR   CAND,,1,SB,WRKU,LAD    CONT WRTP     ; LAD:=WRKU:=SB(0:22) CON 1 EXT U, WRTP
13 CC320 4631000000130324           JMP    F,NWADR,RSREG                        ; IF W-ADDR THEN GOTO RS-REG
14 CC321 0645523500001000           RA     SUBO,ICS,WRKU                        ;
15 CC322 0631000000C20325           JMP    F,NZ,LOAIR                           ; IF WRKU=IC THEN GOTO LOAD-IR
16 CC323 3455730000710100           XZ     OR,CIOST               NEXT TIOIN 1  ; TEST IO, TEST INT
17                                                                              ; GOTO NEXT INSTR
18                          RSREG:                                              ;RSREG:
19 CC324 3457430003600000           XZAB   CDATI,OR,W,LAX         NEXT TSTIN 1  ; LAX:=W, TEST INT
20                                                                              ; GOTO NEXT INSTR
21
22                          LOAIR:                                              ;LOAD-IR:
23 CC325 0655730000510100           XZ     OR,CIOST               CONT ISTIO 1  ; TEST IO
24 CC326 4605730000241011           RZR    OR,ICS,LAD             CONT READJ    ; LAD:=ICS, READ INSTRUCTION
25 CC327 7545230000600000           ZQ     OR                     NEXT TSTIN 3  ; TEST INT
26                                                                              ; GOTO NEXT INSTR
```

```
10016 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;STORE DOUBLE REGISTER, DS, OPCODE = 67
03
04                          DSE:                                                ;DSE:
05 CC330 4624730000156300           IZQ    OR,0,6714                        ;
06 CC331 0631000002C00031           CALL   F,F,ESCAP                        ; CALL ESCAPE(6714)
07                          DS:                                                 ;DS:
08 CC332 0605430000CC0012           ZAR    OR,W,DATO                        ; DATAOUT:= W
09 CC333 4613551332223711           CABR   CAND,.1,SB,WRK0,LAD    CONT WRTP  ; LAD:=WRK0:=SB(0:22) CON 1 EXT 0, WRTP
10 CC334 4631000000130353           JMP    F,NWADR,DSREG                    ; IF W-ADDR THEN GOTO DS-REG
11 CC335 0645523500CC1000           RA     SUB0,ICS,WRK0                    ; IF WRK0=ICS THEN GOTO DS_LOADIR
12 CC336 4631000000020347           JMP    F,NZ,DSIR                        ;
13 CC337 0655730000510100           XZ     OR,CIOST               CONT TSTIO 1 ; TEST IO
14 CC340 4605430040CC0012           ZAR    OR,WPRE,DATO                     ; DATAOUT:=WPRE
15 CC341 4647113426220011           ABBR   SUN0,.2,SB,LAD         CONT WRTP  ; LAD:=SB:=SB-2, WRTP
16 CC342 0631000000130032           JMP    F,NWADR,INT0                     ; IF W-ADDR THEN GOTO INT0
17 CC343 4613551332003700           CAB    CAND,.1,SB,WRK0                  ; WRK0:= SB(0:22) CON 1 EXT 0
18 CC344 0645523500CC1000           RA     SUB0,ICS,WRK0                    ; IF WRK0=ICS THEN GOTO LOAD_IR
19 CC345 0631000000020325           JMP    F,NZ,LOAIR                       ;
20 CC346 3455730000710100           XZ     OR,CIOST               NEXT TIOIN 1 ; TEST IO, TEST INT
21                                                                          ; GOTO NEXT INSTRUCTION
22                          DSIR:                                               ;
23 CC347 0655730000510100           XZ     OR,CIOST               CONT TSTIO 1 ; TEST IO
24 CC350 4605430040CC0012           ZAR    OR,WPRE,DATO                     ; DATAOUT:= WPRE
25 CC351 4647113426220011           ABBR   SUN0,.2,SB,LAD         CONT WRTP  ; SB:=LAD:= SB-2  , WRTP
26 CC352 4631000000CC0325           JMP    F,F,LOAIR                        ; GOTO LOAD_IR
27
28                          DSREG:                                              ;DSREG:
29 CC353 4657430003CC0000           XZAB   CDATI,OR,W,LAX         CONT 0 1   ; LAX:=W
30 CC354 4607113426CC0001           ABBR   SUN0,.2,SB,LAST                   ; LAST:=SB-2
31 CC355 7407430043600000           ZAB    OR,WPRE,LAX            NEXT TSTIN ; LAX:=WPRE, TEST INT
32                                                                          ; GOTO NEXT INSTR
```

```
10017 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;STORE HALF-REGISTER, HS, OPCODE = 32
03
04                          HSE:                                                ;HSE:
05 C0356 0624730000064100           IZQ    OR,0,3204                        ;
06 C0357 0631000002000031           CALL   F,F,ESCAP                        ; CALL ESCAPE(3204)
07                          HS:                                                 ;HS:
08 C0360 0662730033000200           HZB    OR,HSTOR,W,WRK1                  ; WRK1:= IF ODD THEN
09                                                                          ;   12 EXT 0 CON W(12:23)
10                                                                          ; ELSE
11                                                                          ;   W(12:23) CON 12 EXT 0
12 C0361 4631000000130371           JMP    F,NWADR,HSREG                    ; IF W-ADDR THEN GOTO HS-REG
13 C0362 4654730000514000           XZQ    OR,DATI                CONT TSTIO 1 ; Q:=DATAIN, TEST IO
14 C0363 4662641532000300           HQB    AND,MASK,WRK0,WRK0               ; WRK0:= IF ODD THEN
15                                                                          ;   Q(0:11) CON 12 EXT 0
16                                                                          ; ELSE
17                                                                          ;   12 EXT 0 CON Q(12:23)
18 C0364 4605131533000012           ABR    OR,WRK0,WRK1,DATO                ; DATAOUT:=WRK0 OR WRK1
19 C0365 4613551332223711           CABR   CAND,.1,SB,WRK0,LAD    CONT WRTP     ; LAD:=WRK0:=SB(0:22) CON 1 EXT 0, WRTP
20 C0366 0645523500001000           RA     SUB0,ICS,WRK0                    ; IF IC=WRK0 THEN
21 C0367 0631000000020325           JMP    F,NZ,LOAIR                       ; GOTO LOAD-IR
22 C0370 3455730000710100           XZ     OR,CIOST               NEXT TIOIN 1 ; TEST IO, TEST INT
23                                                                          ; GOTO NEXT INSTR
24                          HSREG:                                              ;HSREG:
25 C0371 4604430140000000           ZAQ    OR,LAX                           ; Q:= LAX
26 C0372 4662640143000300           HQB    AND,MASK,LAX,LAX                 ; LAX:= IF ODD THEN
27                                                                          ;   Q(0:11) CON 12 EXT 0
28                                                                          ; ELSE
29                                                                          ;   12 EXT 0 CON Q(12:23)
30 C0373 3457131543600000           XABB   CDATI,OR,WRK1,LAX      NEXT TSTIN 1 ; LAX:=LAX OR WRK1, TEST INT
31                                                                          ; GOTO NEXT INSTR
```

```
10018 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;STORE EXCEPTION REGISTER, XS, OPCODE = 33
03
04                          XSE:                                                ;XSE:
05 CC374 4624730000066100           IZQ   OR,0,3304                           ;
06 CC375 0631000002CC0031           CALL  F,F,ESCAP                           ; CALL ESCAPE(3304)
07                          XS:                                                 ;XS:
08 CC376 0613541233CC3400           CAB   AND,.7,STAT,WRK1                    ; WRK1:= 21 EXT 0 CON STAT(21:23)
09 CC377 0662731573CC0200           HZB   OR,HSTOR,WRK1,WRK1                  ; WRK1:= IF ODD THEN
10                                                                            ;   12 EXT 0 CON WRK1(12:23)
11                                                                            ; ELSE
12                                                                            ;   WRK1(12:23) CON 12 EXT 0
13 CC400 4631000000130371           JMP   F,NWADR,HSREG                       ; IF W=ADDR THEN GOTO HS-REG
14 CC401 4654730000514000           XZQ   OR,DATI                 CONT TSTIO 1 ; Q:=DATAIN, TEST IO
15 CC402 4662641532CC0300           HQB   AND,MASK,WRK0,WRK0                  ; WRK0:= IF ODD THEN
16                                                                            ;   Q(0:11) CON 12 EXT 0
17                                                                            ; ELSE
18                                                                            ;   12 EXT 0 CON Q(12:23)
19 CC403 0607131533CC0012           ABBR  OR,WRK0,WRK1,DATO                   ; DATAOUT:= WRK1 OR WRK0
20 CC404 4613551332223711           CABR  CAND,.1,SB,WRK0,LAD     CONT WRTP   ; LAD:=WRK0:=SB(0:22) CON 1 EXT 0, WRTP
21 CC405 0645523500CC1000           RA    SUB0,ICS,WRK0                       ; IF ICS=WRK0 THEN
22 CC406 0631000000020325           JMP   F,NZ,LOAIR                          ; GOTO LOAD-IR
23 CC407 3455730000710100           XZ    OR,CIOST                NEXT TIOIN 1 ; TEST IO, TEST INT
24                                                                            ; GOTO NEXT INSTR
25
26                          ;EXCHANGE REGISTER AND MEMORY WORD, RX, OPCODE = 31
27
28                          RXE:                                                ;RXE:
29 CC410 4624730000062140           IZQ   OR,0,3106                           ;
30 CC411 0631000002CC0031           CALL  F,F,ESCAP                           ; CALL ESCAPE(3106)
31                          RX:                                                 ;RX:
32 CC412 0631000000130424           JMP   F,NWADR,RXREG                       ; IF W=ADDR THEN GOTO RX-REG
33 CC413 0605430000CC0012           ZAR   OR,W,DATO                           ; DATAOUT:=W
34 CC414 4657730033514000           XZB   OR,DATI,WRK1            CONT TSTIO 1 ; WRK1:=DATAIN, TEST IO
35 CC415 4613551332223711           CABR  CAND,.1,SB,WRK0,LAD     CONT WRTP   ; LAD:=WRK0:=SB(0:22) CON 1 EXT 0, WRTP
36 CC416 0645523500001000           RA    SUB0,ICS,WRK0                       ;
```

```
10019 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 CC417 0631000000020421          JMP   F,NZ,RXLOA                          ; IF IC<>WRKU THEN
03                                                                           ; BEGIN
04 CC420 7457431540710100          XZAB  CIOST,OR,WRK1,W       NEXT TIOIN 1 ;  W:= WRK1,  TEST IO,  TEST INT
05                                                                           ;   GOTO NEXT INSTR
06                                                                           ; END
07 CC421 4657431540510100 RXLOA:   XZAB  CIOST,OR,WRK1,W       CONT TSTIO 1 ; W:= WRK1,  TEST IO
08 CC422 4605730000241011          RZR   OR,ICS,LAD            CONT READJ   ;  LAD:= ICS ,  READ INSTRUCTION
09 CC423 7545230000600000          ZQ    OR                    NEXT TSTIN 3 ; TEST INT
10                                                                           ;  GOTO NEXT INSTRUCTION
11                                 RXREG:                                    ;RXREG:
12 CC424 0657430172CC0000          XZAB  CDATI,OR,LAX,WRKO     CONT 0 1     ; WRKO:=LAX
13 CC425 4607430003CC0000          ZAB   OR,W,LAX                           ; LAX:=W
14 CC426 7407431500600000          ZAB   OR,WRKU,W             NEXT TSTIN   ; W:=WRKU, TEST INT
15                                                                           ; GOTO NEXT INSTR
16
17                                 ;DOUBLEWORD INSTRUCTIONS WITH MEMORY REFERENCE
18                                 ;*********************************************
19
20                                 ;LOAD DOUBLE REGISTER, DL, OPCODE = 66
21
22                                 DLE:                                      ;DLE:
23 CC427 0624730000154240          1ZQ   OR,0,6612                          ;
24 CC430 0631000002000031          CALL  F,F,ESCAP                          ; CALL ESCAPE(6612)
25                                 DL:                                       ;DL:
26 CC431 0631000000130436          JMP   F,NWADR,DLREG                      ; IF W-ADDR THEN GOTO DL-REG
27 CC432 4657730000514000          XZB   OR,DATI,W             CONT TSTIO 1 ; W:=DATAIN, TEST IO
28 CC433 0647113426200011          ABBR  SUNO,,2,SB,LAD        CONT READP   ; LAD:=SB:=SB-2, READP
29 CC434 0631000000130032          JMP   F,NWADR,INTU                       ; IF W-ADDR THEN GOTO INTU
30 CC435 3457730001714000          XZB   OR,DATI,WPRE          NEXT TIOIN 1 ; WPRE:=DATAIN, TEST IO,TEST INT
31                                                                           ; GOTO NEXT INSTR
32                                 DLREG:                                    ;DLREG:
33 CC436 4657430140000000          XZAB  CDATI,OR,LAX,W        CONT 0 1     ; W:=LAX
34 CC437 4607113426CC0001          AEBR  SUNO,,2,SB,LAST                    ; SB:=LAST:=SB-2
35 CC440 7407430141600000          ZAB   OR,LAX,WPRE           NEXT TSTIN   ; WPRE:=LAX, TEST INT
36                                                                           ; GOTO NEXT INSTR
```

```
10020 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;ADD INTEGER DOUBLE WORD, AA, OPCODE = 7U
03
04                          AAE:                                                    ;AAE:
05 CC441 4624730000160240           IZQ    OR,0,7012                            ;
06 CC442 0631000002CC0031           CALL   F,F,ESCAP                            ; CALL ESCAPE(7012)
07                          AA:                                                     ;AA:
08 CC443 0631000000130453           JMP    F,NWADR,AAREG                        ; IF W=ADDR THEN GOTO AA=REG
09 CC444 4657730033514000           XZB    OR,DATI,WRK1          CONT TSTIO 1   ; WRK1:=DATAIN, TEST IO
10 CC445 0647113426200011           ABBR   SUNO,,2,SB,LAD        CONT READP     ; SB:=LAD:=SB-2, READP
11 CC446 0631000000130032           JMP    F,NWADR,INTU                         ; IF W=ADDR THEN GOTO INTO
12 CC447 0657730032514000           XZB    OR,DATI,WRKU          CONT TSTIO 1   ; WRKU:=DATAIN, TEST IO
13 CC450 4607101540CC0000           ABB    ADD,WRK1,W                           ; W:=W + WRK1
14 CC451 0607105501CC0000           ABB    ADDC,WRKU,WPRE                       ; WPRE:=WPRE + WRKU + CARRY
15 CC452 7406731224644000           EZB    OR,STOC,STAT,STAT     NEXT TOVIN     ; EX(22:23):=OVERFLOW CON CARRY
16                                                                               ; TEST OVERFLOW, TEST INT
17                                                                               ; GOTO NEXT INSTR
18                          AAREG:                                                  ;AAREG:
19 CC453 4657430173CC0000           XZAB   CDATI,OR,LAX,WRK1     CONT U 1       ; WRK1:=LAX
20 CC454 4607113426CC0001           ABBR   SUNO,,2,SB,LAST                       ; LAST:=SB:=SB-2
21 CC455 4607101540CC0000           ABB    ADD,WRK1,W                           ; W:=W + WRK1
22 CC456 4607104141CC0000           ABB    ADDC,LAX,WPRE                        ; WPRE:=WPRE + LAX + CARRY
23 CC457 7406731224644000           EZB    OR,STOC,STAT,STAT     NEXT TOVIN     ; EX(22:23):= OVERFLOW CON CARRY
24                                                                               ; TEST OVERFLOW, TEST INT
25                                                                               ; GOTO NEXT INSTR
```

```
10021 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;SUBTRACT INTEGER DOUBLE WORD, SS, OPCODE = 71
03
04                          SSE:                                                ;SSE:
05 CC460 0624730000162240           IZQ    OR,0,7112                        ;
06 CC461 0631000002CC0031           CALL   F,F,ESCAP                        ; CALL ESCAPE(7112)
07                          SS:                                                 ;SS:
08 CC462 0631000000130472           JMP    F,NWADR,SSREG                    ; IF W=ADDR THEN GOTO SS=REG
09 CC463 4657730033514000           XZB    OR,DATI,WRK1          CONT TSTIO 1 ; WRK1:=DATAIN, TEST IO
10 CC464 0647113426200011           ABBR   SUNO,,2,SB,LAD        CONT READP   ; SB:=LAD:=SB-2, READP
11 CC465 0631000000130032           JMP    F,NWADR,INTU                     ; IF W=ADDR THEN GOTO INTU
12 CC466 0657730032514000           XZB    OR,DATI,WRK0          CONT TSTIO 1 ; WRK0:=DATAIN, TEST IO
13 CC467 4607113540CC0000           ABB    SUNO,WRK1,W                      ; W:=W - WRK1
14 CC470 4607115501C00000           ABB    SUNC,WRK0,WPRE                   ; WPRE:=WPRE - WRK0 - 1 + CARRY
15 CC471 7406731224644000           EZB    OR,STOC,STAT,STAT     NEXT TOVIN   ; EX(22:23):= OVERFLOW CON CARRY
16                                                                          ; TEST OVERFLOW, TEST INT
17                                                                          ; GOTO NEXT INSTRUCTION
18                          SSREG:                                              ;SSREG:
19 CC472 4657430173CC0000           XZAB   CDATI,OR,LAX,WRK1     CONT 0 1     ; WRK1:=LAX
20 CC473 4607113426CC0001           ABBR   SUNO,,2,SB,LAST                    ; SB:=LAST:=SB-2
21 CC474 4607113540CC0000           ABB    SUNO,WRK1,W                      ; W:=W - WRK1
22 CC475 0607114141000000           ABB    SUNC,LAX,WPRE                    ; WPRE:=WPRE - LAX - 1 + CARRY
23 CC476 7406731224644000           EZB    OR,STOC,STAT,STAT     NEXT TOVIN   ; EX(22:23):=OVERFLOW CON CARRY
24                                                                          ; TEST OVERFLOW, TEST INT
25                                                                          ; GOTO NEXT INSTR
```

```
10022 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;SHIFT INSTRUCTIONS
03                          ;******************
04
05                          ;ARITHMETIC SHIFT SINGLE, AS, OPCODE = 44
06
07                          ASE:                                                ;ASE:
08 CC477 4624730000110020           IZQ    OR,0,4401                        ;
09 CC500 0631000002CC0031           CALL   F,F,ESCAP                        ; CALL ESCAPE(4401)
10                          AS:                                                 ;AS:
11 CC501 0631000000410506           JMP    T,NEG,ASRGT                      ; IF SB<0 THEN GOTO AS-RIGHT
12 CC502 0631000000020510           JMP    F,NZ,ASNO                        ; IF SB=0 THEN GOTO AS-NO
13 CC503 0251523300CC3704           CAR    SUBO,.1,SB,LCD          PUSH     ; LCD:= -SB+1 , COUNT:= MIN(SB,48)
14 CC504 2637330000520060           SLZBB  OR,W,Z,NL,T,MAXLO       LRTN     ; W:= W(COUNT:23+COUNT)
15 CC505 3406731224624100           EZB    OR,STSZ,STAT,STAT       NEXT TSHIN ; EX:= SHIFT OFL, ZERO
16                                                                          ; TEST SHIFT OFL, TEST INT,
17                                                                          ; GOTO NEXT INSTRUCTION
18                          ASRGT:                                              ;AS-RIGHT:
19 CC506 0205403300CC0004           ZAR    ADDO,SB,LCD             PUSH     ; LCD:=SB+1 , PUSH
20                                                                          ; COUNT:= MAX(SB,-48)
21 CC507 2635330000522060           SRZBB  OR,W,FO,NL,T,MAXLO      LRTN     ; W:= SIGN EXTENDED W(0:23+COUNT)
22                          ASNO:                                               ;AS-NO:
23 CC510 7406731224604200           EZB    OR,STZZ,STAT,STAT       NEXT TSTIN ; EX:= 0,0,  TEST INT
24                                                                          ; GOTO NEXT INSTRUCTION
25
26                          ;ARITHMETIC SHIFT DOUBLE, AD, OPCODE = 45
27
28                          ADE:                                                ;ADE:
29 CC511 0624730000112020           IZQ    OR,0,4501                        ;
30 CC512 0631000002CC0031           CALL   F,F,ESCAP                        ; CALL ESCAPE(4501)
31                          AD:                                                 ;AD:
32 CC513 0631000000410524           JMP    T,NEG,ADRGT                      ; IF SB<0 THEN GOTO AD-RIGHT
33 CC514 4631000000C20530           JMP    F,NZ,ADNO                        ; IF SB=0 THEN GOTO AD-NO
34 CC515 4604430000CC0000           ZAQ    OR,W                             ; Q:=W
35 CC516 0251523300CC3704           CAR    SUBO,.1,SB,LCD          PUSH     ; LCD:= -SB+1 , PUSH
36                                                                          ; COUNT:= MIN(48,SB)
37 CC517 2636330001520060           DLZBB  OR,WPRE,Z,NL,T,MAXLO    LRTN     ; WPRE,Q:= WPRE,Q (COUNT:47+COUNT)
38 CC520 0606731224004100           EZB    OR,STSZ,STAT,STAT                ; EX:= SHIFT OFL, ZERO
```

```
10C23 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 C0521 0613541232001400          CAB   AND,.3,STAT,WRKC                    ; WRK0:= SHIFT OFL, ZERO
03 C0522 4651523500003500          CA    SUB0,.1B0,WRK0                      ; IF WRK0<>0 THEN SET OVERFLOW
04 C0523 7407230000640000          ZQB   OR,W                      NEXT TOVIN ; W:= Q , TEST OVERFLOW
05                                                                            ; GOTO NEXT INSTR
06                          ADRGT:                                            ;AD-RIGHT:
07 C0524 4604430000C00000          ZAQ   OR,W                                ; Q:=W
08 C0525 0205403300000004          ZAR   ADD0,SB,LCD               PUSH      ; LCD:= SB+1 , PUSH
09                                                                            ; COUNT:= MAX(SB,-48)
10 C0526 2634330001522060          DRZBB OR,WPRE,F0,NL,T,MAXLO     LRTN      ; WPRE,Q := SIGN EXT WPRE,Q(0:47+COUNT)
11 C0527 4607230000000000          ZQB   OR,W                                ; W:= Q
12                          ADNO:                                             ;AD-NO:
13 C0530 3406731224624200          EZB   OR,STZZ,STAT,STAT         NEXT TSHIN ; EX:= 0,0 , TEST INT
14                                                                            ; GOTO NEXT INSTR
15
16                          ;LOGICAL SHIFT SINGLE, LS, OPCODE = 46
17
18                          LSE:                                              ;LSE:
19 C0531 0624730000114020          1ZQ   OR,0,4601                           ;
20 C0532 0631000002000031          CALL  F,F,ESCAP                           ; CALL ESCAPE(4601)
21                          LS:                                               ;LS:
22 C0533 4631000000410540          JMP   T,NEG,LSRGT                         ; IF SB<0 THEN GOTO LS-RIGHT
23 C0534 0631000000020537          JMP   F,NZ,LSNO                           ; IF SB=0 THEN GOTO LS-NO
24 C0535 0251523300003704          CAR   SUB0,.1,SB,LCD            PUSH      ; LCD:= -SB+1 , PUSH
25                                                                            ; COUNT:= MIN(SB,48)
26 C0536 2637330000520060          SLZBB OR,W,Z,NL,T,MAXLO         LRTN      ; W:= W(COUNT:23+COUNT)
27 C0537 7407330000600000 LSNO:    ZBB   OR,W                      NEXT TSTIN ; TEST INT
28                                                                            ; GOTO NEXT INSTR
29                          LSRGT:                                            ;LS-RIGHT:
30 C0540 0205403300000004          ZAR   ADD0,SB,LCD               PUSH      ; LCD:= SB+1, PUSH
31                                                                            ; COUNT:= MAX(SB,-48)
32 C0541 6635330000520060          SRZBB OR,W,Z,NL,T,MAXLO         LRTN      ; W:= W(COUNT:23+COUNT)
33 C0542 7407330000600000          ZBB   OR,W                      NEXT TSTIN ; TEST INT
34                                                                            ; GOTO NEXT INSTRUCTION
```

```
10024 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;LOGICAL SHIFT DOUBLE, LD, OPCODE = 47
03
04                          LDE:                                                ;LDE:
05 CC543 4624730000116020           IZQ    OR,0,4701                        ;
06 CC544 0631000002CC0031           CALL   F,F,ESCAP                        ; CALL ESCAPE(4701)
07                          LD:                                                 ;LD:
08 CC545 4631000000410554           JMP    T,NEG,LDRGT                      ; IF SB<0 THEN GOTO LD-RIGHT
09 CC546 4631000000020553           JMP    F,NZ,LDNO                        ; IF SB=0 THEN GOTO LD-NO
10 CC547 4604430000CC0000           ZAQ    OR,W                             ; Q:=W
11 CC550 0251523300CC3704           CAR    SUBO,.1,SB,LCD          PUSH     ; LCD:= -SB+1  , PUSH
12                                                                          ; COUNT:= MIN(SB,48)
13 CC551 2636330001520060           DLZBB  OR,WPRE,Z,NL,T,MAXLO    LRTN     ; WPRE,Q:= WPRE,Q(COUNT:47+COUNT)
14 CC552 3407230000600000           ZQB    OR,W                     NEXT TSTIN ; W:=Q, TEST INT
15                                                                          ; GOTO NEXT INSTR
16 CC553 7405230000600000 LDNO:     ZQ     OR                       NEXT TSTIN ;LD-NO: TEST INT,
17                                                                          ; GOTO NEXT INSTR
18                          LDRGT:                                              ;LD-RIGHT:
19 CC554 4604430000CC0000           ZAQ    OR,W                             ; Q:= W
20 CC555 0205403300CC0004           ZAR    ADDO,SB,LCD             PUSH     ; LCD:= SB+1 , PUSH
21                                                                          ; COUNT:= MAX(SB,-48)
22 CC556 6634330001520060           DRZBB  OR,WPRE,Z,NL,T,MAXLO    LRTN     ; WPRE,Q:= WPRE,Q(SB:47+SB)
23 CC557 3407230000600000           ZQB    OR,W                     NEXT TSTIN ; W:=Q, TEST INT
24                                                                          ; GOTO NEXT INSTR
```

```
10025 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;NORMALIZE SINGLE, NS, OPCODE = 42
03
04                          NSE:                                                ;NSE:
05 CC560 4624730000104100           IZQ    OR,0,4204                        ;
06 CC561 0631000002CC0031           CALL   F,F,ESCAP                        ; CALL ESCAPE(4204)
07                          NS:                                                 ;NS:
08 CC562 0605240CCCCC0004           ZQR    AND,LCD                          ; LCD:=0
09 CC563 0637330000CC0060           SLZBB  OR,W,Z,NL,F,F                    ; W:= W SHIFT 1
10 CC564 0631000CCCC20602           JMP    F,NZ,NSZER                       ; IF W = 0 THEN GOTO NS-ZERO
11 CC565 4231000000460567           JMP    T,NORM0,NS0          PUSH        ; IF W(0)=W(1) THEN GOTO NS0
12 CC566 6637330000470060           SLZBB  OR,W,Z,NL,T,NORM1    LRTN        ; REPEAT
13 CC567 0635330000001060 NS0:      SRZBB  OR,W,LNK,NL,F,F                  ;   W:= W SHIFT 1
14                                                                          ; UNTIL W(0)<>W(1)
15 CC570 0647712033003000 NS00:     RZB    SUN0,LCS,WRK1                    ; WRK1:= -LCS
16 CC571 0662731573CC0200           HZB    OR,HSTOR,WRK1,WRK1               ; WRK1:= IF ODD THEN
17                                                                          ;   12 EXT 0 CON WRK1(12:23)
18                                                                          ; ELSE
19                                                                          ;   WRK1(12:23) CON 12 EXT 0
20 CC572 4631000000130371 NS1:      JMP    F,NWADR,HSREG                    ; IF W-ADDR THEN GOTO HS-REG
21 CC573 4654730000514000           XZQ    OR,DATI              CONT TSTIO 1 ; Q:= DATAIN, TEST IO
22 CC574 4662641532CC0300           HQB    AND,MASK,WRK0,WRK0               ; WRK0:= IF ODD THEN
23                                                                          ;   Q(0:11) CON 12 EXT 0
24                                                                          ; ELSE
25                                                                          ;   12 EXT 0 CON Q(12:23)
26 CC575 4605131533000012           ABR    OR,WRK0,WRK1,DATO                ; DATAOUT:= WRK0 OR WRK1
27 CC576 4613551332223711           CABR   CAND,.1,SB,WRK0,LAD  CONT WRTP   ; LAD:=WRK0:=SB(0:22) CON 1 EXT 0, WRTP
28 CC577 0645523500CC1000           RA     SUB0,ICS,WRK0                    ; IF ICS=WRK0 THEN
29 CC600 0631000000020325           JMP    F,NZ,LOAIR                       ;   GOTO LOAD-IR
30 CC601 3455730000710100           XZ     OR,CIOST             NEXT TIOIN 1 ; TEST IO, TEST INT
31                                                                          ; GOTO NEXT INSTR
32                          NSZER:                                              ;NSZER:
33 CC602 0613730033002300           CZB    OR,.1B12,WRK1                    ; WRK1:= 8'4000
34 CC603 0631000000CC0572           JMP    F,F,NS1                          ; GOTO NS1
```

```
10026 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;NORMALIZE DOUBLE, ND, OPCODE = 43
03
04                          NDE:                                                ;NDE:
05 CC604 0624730000106100           IZQ    OR,0,4304                        ;
06 CC605 0631000002000031           CALL   F,F,ESCAP                        ; CALL ESCAPE(4304)
07                          ND:                                                 ;ND:
08 CC606 0605330001000000           ZB     OR,WPRE                          ; IF WPRE=0 AND W=0 THEN
09 CC607 0631000000420612           JMP    T,NZ,ND1                         ;   GOTO NS=ZERO
10 CC610 4605330000000000           ZB     OR,W                             ;
11 CC611 0631000000020602           JMP    F,NZ,NSZER                       ;
12 CC612 0604330000000000 ND1:      ZBQ    OR,W                             ; Q:=W
13 CC613 0605240000000004           ZQR    AND,LCD                          ; LCD:= 0
14 CC614 0636330001000060           DLZBB  OR,WPRE,Z,NL,F,F                 ; IF WPRE(0)=WPRE(1) THEN
15 CC615 0231000000460617           JMP    T,NORMU,NDO              PUSH    ;   GOTO NDU ELSE PUSH
16 CC616 6636330001470060           DLZBB  OR,WPRE,Z,NL,T,NORM1     LRTN    ; REPEAT
17 CC617 0634330001001060 NDO:      DRZBB  OR,WPRE,LNK,NL,F,F               ;   WPRE,Q:= WPRE,Q SHIFT 1
18                                                                          ; UNTIL WPRE(0)<>WPRE(1)
19 CC620 4607230000000000           ZQB    OR,W                             ; W:=Q
20 CC621 4631000000000570           JMP    F,F,NSOO                         ; GOTO NSOU
```

```
10027 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;JUMP INSTRUCTIONS
03                          ;******************
04
05                          ;JUMP WITH REGISTER LINK, JL, OPCODE = 15
06                          ;NO LINK
07
08                          JLE:                                                ;JLE:
09 C0622 0624730000032400           IZQ    OR,0,1520                            ;
10 C0623 0631000002C00031           CALL   F,F,ESCAP                            ; CALL ESCAPE(1520)
11                          JL:                                                 ;JL:
12 C0624 0631000000130032           JMP    F,NWADR,INTO                         ; IF W-ADDR THEN GOTO INTO
13 C0625 0655431300410103           XZAR   CIOST,OR,SB,ICD       CONT TSTLI 1   ; TEST LIMITVIOLATION, ICD:=SB
14 C0626 7405230000600000           ZQ     OR                    NEXT TSTIN     ; TEST INT, GOTO NEXT INSTR
15
16                          ;WITH LINK
17
18                          JLLE:                                               ;JLLE:
19 C0627 0624730000032400           IZQ    OR,0,1520                            ;
20 C0630 0631000002C00031           CALL   F,F,ESCAP                            ; CALL ESCAPE(1520)
21                          JLL:                                                ;JLL:
22 C0631 0631000000130032           JMP    F,NWADR,INTO                         ; IF W-ADDR THEN GOTO INTO
23 C0632 0604730000001000           RZQ    OR,ICS                               ; Q:= IC
24 C0633 0655431300410103           XZAR   CIOST,OR,SB,ICD       CONT TSTLI 1   ; TEST LIMITVIOLATION, ICD:= SB
25                                                                              ; W:= Q
26 C0634 3407230000600000           ZQB    OR,W                  NEXT TSTIN     ; TEST INT, GOTO NEXT INSTR
```

```
!0028 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;JUMP WITH INTERRUPT DISABLED, JD, OPCODE = 16
03                          ;NO LINK
04
05                          JDE:                                                ;JDE:
06 CC635 4624730000035400           IZQ   OR,0,1660                           ;
07 CC636 0631000002CC0031           CALL  F,F,ESCAP                           ; CALL ESCAPE(1660)
08                          JD:                                                 ;JD:
09 CC637 0641501300CC0600           SA    ADD,MONT,SB                         ; IF SB>= -MONTOP THEN
10 CC640 0631000000C10643           JMP   F,NEG,DSABL                         ;   GOTO DISABLE
11 CC641 0651501300CC2300           CA    ADD,.1B12,SB                        ; IF SB >= -8'4000 THEN
12 CC642 0631000000C10651           JMP   F,NEG,MCALL                         ;   GOTO MONCALL
13                          DSABL:                                              ;DISABLE:
14 CC643 0631000000130032           JMP   F,NWADR,INTO                        ; IF W=ADDR THEN GOTO INTO
15 CC644 0655431300410103           XZAR  CIOST,OR,SB,ICD       CONT TSTLI 1  ; TEST LIMITVIOLATION, ICD:= SB
16 CC645 0613531224003605           CABR  OR,.1B20,STAT,STAT,CPUST            ; STAT:=CPUST:= STAT OR BIT 20 (DISABLE)
17 CC646 0745230000CC0000           ZQ    OR                    CONT U 3      ; DELAY 495 NSEC FOR CHANGE OF LEVEL
18 CC647 4705230000CC0000 DSAB1:    ZQ    OR                    CONT U 2      ;
19 CC650 7405230000C6C0000          ZQ    OR                    NEXT TSTIN    ; TEST INT, GOTO NEXT INSTR
20                          MCALL:                                              ;MONCALL:
21 CC651 0613551471CC1706           CABR  CAND,.1B18,CNTR,CNTR,CNTRO          ; AUTOLOADING:=0
22 CC652 0607411367000000           ZAB   SUN,INF,INF                         ; INF:=INF-1
23 CC653 0653501325CC2300           CAB   ADD,.1B12,SB,CAUSE                  ; CAUSE:= SB+ 8'4000, MON CALL NO
24 CC654 4631000000001434           JMP   F,F,ICALL                           ; GOTO INTERRUPT CALL
25
26                          ;WITH LINK
27
28                          JDLE:                                               ;JDLE:
29 CC655 4624730000035400           IZQ   OR,0,1660                           ;
30 CC656 0631000002CC0031           CALL  F,F,ESCAP                           ; CALL ESCAPE(1660)
31                          JDL:                                                ;JDL:
32 CC657 0641501300CC0600           SA    ADD,MONT,SB                         ; IF SB>= -MONTOP THEN
33 CC660 4631000000010663           JMP   F,NEG,DSALL                         ;   GOTO DISABLE_LINK
34 CC661 0651501300CC2300           CA    ADD,.1B12,SB                        ; IF SB>= -8'4000 THEN
35 CC662 0631000000010651           JMP   F,NEG,MCALL                         ;   GOTO MONCALL
```

```
10029 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          DSALL:                                                  ;DISABLE_LINK:
03 CC663 0631000000130032           JMP   F,NWADR,INTO                              ; IF W_ADDR THEN GOTO INTO
04 CC664 0604730000001000           RZQ   OR,ICS                                    ; Q:= IC
05 CC665 0655431300410103           XZAR  CIOST,OR,SB,ICD       CONT TSTLI 1        ;TEST LIMIT, IC:= SB
06 CC666 0613531224003605           CABR  OR,.1B20,STAT,STAT,CPUST                  ; STAT:=CPUST:= STAT OR BIT20 (DISABLE)
07 CC667 0745230000000000           ZQ    OR                    CONT 0 3            ; DELAY 495 NSEC
08 CC670 4705230000000000 DSAL1:    ZQ    OR                    CONT 0 2            ;
09 CC671 3407230000000000           ZQB   OR,W                  NEXT TSTIN          ; W:= Q, TEST INT, GOTO NEXT INSTR
10
11                          ;JUMP WITH INTERRUPT ENABLED, JE, OPCODE = 17
12                          ;NO LINK
13
14                          JEE:                                                    ;JEE:
15 CC672 0624730000037400           IZQ   OR,0,1760                                 ;
16 CC673 0631000002000031           CALL  F,F,ESCAP                                 ; CALL ESCAPE(1760)
17                          JE:                                                     ;JE:
18 CC674 0631000000130032           JMP   F,NWADR,INTO                              ; IF W=ADDR THEN GOTO INTO
19 CC675 0655431300410103           XZAR  CIOST,OR,SB,ICD       CONT TSTLI 1        ; TESTLIMITVIOLATION, ICD:= SB
20 CC676 0613551224003605           CABR  CAND,.1B20,STAT,STAT,CPUST                ; STAT:=CPUST:= STAT CAND BIT 20 (ENABLE)
21 CC677 0631000000000647           JMP   F,F,DSAB1                                 ; GOTO DSAB1
22
23                          ;JUMP WITH INTERRUPT ENABLED, JE, OPCODE = 17
24                          ;WITH LINK
25
26                          JELE:                                                   ;JELE:
27 CC700 0624730000037400           IZQ   OR,0,1760                                 ;
28 CC701 0631000002000031           CALL  F,F,ESCAP                                 ; CALL ESCAPE(1760)
29                          JEL:                                                    ;JEL:
30 CC702 0631000000130032           JMP   F,NWADR,INTO                              ; IF W=ADDR THEN GOTO INTO
31 CC703 0604730000001000           RZQ   OR,ICS                                    ; Q:= ICS
32 CC704 0655431300410103           XZAR  CIOST,OR,SB,ICD       CONT TSTLI 1        ; TESTLIMITVIOLATION, ICD:= SB
33 CC705 0613551224003605           CABR  CAND,.1B20,STAT,STAT,CPUST                ; STAT:=CPUST:= STAT CAND BIT 20 (ENABLE)
34 CC706 4631000000000670           JMP   F,F,DSAL1                                 ; GOTO DSAL1
```

```
10030 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;FLOATING POINT INSTRUCTIONS
03                          ;**************************
04
05                          ;CONVERT INTEGER TO FLOATING, CI, OPCODE = 40
06
07                          CIE:                                                ;CIE:
08 C0707 0624730000100020           IZQ    OR,0,4001                        ;
09 C0710 0631000002000031           CALL   F,F,ESCAP                        ; CALL ESCAPE(4001)
10                          CI:                                                 ;CI:
11 C0711 0607430001000000           ZAB    OR,W,WPRE                        ; WPRE:=W
12 C0712 4651712000000404           CZR    SUN0,.23,LCD                     ; LCD:= -SB -23
13 C0713 0645523300003004           RAR    SUB0,LCS,SB,LCD                  ;
14 C0714 0604240000000000           ZQQ    AND                              ; Q:= 0
15                          NORMA:                                              ;NORMALIZE:
16                                                                          ; WPRE,Q= FRACTION
17                                                                          ; LC= -EXPONENT
18 C0715 0605330001000000           ZB     OR,WPRE                          ; IF WPRE=0 AND W=0 THEN
19 C0716 0631000000420723           JMP    T,NZ,CINZ                        ; BEGIN
20 C0717 0605230000000000           ZQ     OR                               ;   W:=8'4000 , TEST INT
21 C0720 0631000000420723           JMP    T,NZ,CINZ                        ;   GOTO NEXT INSTR
22 C0721 0613730000002300 CIZER:    CZB    OR,.1B12,W                       ;   EX(22:23):= 0
23 C0722 7406731224604200           EZB    OR,STZZ,STAT,STAT      NEXT TSTIN ; END
24                          CINZ:                                               ;
25 C0723 0636330001000060           DLZBB  OR,WPRE,Z,NL,F,F                 ;
26 C0724 0231000000460726           JMP    T,NORM0,CI0           PUSH       ; IF WPRE(0)=WPRE(1) THEN
27 C0725 6636330001470060           DLZBB  OR,WPRE,Z,NL,T,NORM1  LRTN       ;   NORMALIZE WPRE,Q
28 C0726 0634330001001060 CI0:      DRZBB  OR,WPRE,LNK,NL,F,F               ;
29 C0727 0650600000002300           CQQ    ADD,.1B12                        ; Q:= Q+ 8'4000
30                          ROUND:                                              ;ROUND:
31 C0730 4674304001003060           DRZBB  ADDC,WPRE,SGN,NL,F,F             ; WPRE,Q:= (WPRE+CARRY,Q) SHIFT 1
32 C0731 0636330001000060           DLZBB  OR,WPRE,Z,NL,F,F                 ;
33 C0732 0231000000460734           JMP    T,NORM0,CI00          PUSH       ; IF WPRE(0)=WPRE(1) THEN
34 C0733 6636330001470060           DLZBB  OR,WPRE,Z,NL,T,NORM1  LRTN       ;   NORMALIZE WPRE,Q
35 C0734 0634330001001060 CI00:     DRZBB  OR,WPRE,LNK,NL,F,F               ;
36 C0735 0610650000000300           CQQ    CAND,B12                         ; Q:= Q(0:11) CON 12 EXT 0
```

```
10031 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 CC736 4653702032CC2300          CZB   ADDO,.1B12,WRK0                    ; WRK0:= -LCS + 8'4001
03 CC737 4647513532CC3000          RAB   SUNO,LCS,WRK0,WRK0                 ;
04 CC740 4613561532CC2300          CAB   EXOR,.1B12,WRK0,WRK0               ; WRK0:= WRK0 EXOR 8'4000
05 CC741 0613541500000300          CAB   AND,B12,WRK0,W                     ; W:= 12 EXT 0 CON WRK0(12:23)
06 CC742 0607030000CC0000          AQB   OR,W,W                             ; W:= W OR Q
07 CC743 0606731224CC4200          EZB   OR,STZZ,STAT,STAT                  ; EX(22:23):= 0
08 CC744 0611551500CC0300          CA    CAND,B12,WRK0                      ; IF WRK0(0:11)=0 THEN
09 CC745 4631000000420747          JMP   T,NZ,CI1                           ;   TEST INT, GOTO NEXT INSTR
10 CC746 7405230000600000          ZQ    OR                    NEXT TSTIN   ;
11                          CI1:                                            ;
12 CC747 0631000000410751          JMP   T,NEG,CIUFL                        ; IF WRK0<0 THEN EX(22:23):= 1
13 CC750 4607302024CC0000          ZBB   ADDO,STAT                          ; ELSE EX(22:23):= 2
14 CC751 4607302024CC0000 CIUFL:   ZBB   ADDO,STAT                          ; IF FP-MASK THEN
15 CC752 4611541200CC2600 TINTF:   CA    AND,.1B5,STAT                      ;   GOTO INTF
16 CC753 0631000000420035          JMP   T,NZ,INTF                          ;
17 CC754 7405230000600000          ZQ    OR                    NEXT TSTIN   ; TEST INT, GOTO NEXT INSTR
18
19                          ;CONVERT FLOATING TO INTEGER, CF, OPCODE = 65
20
21                          CFE:                                            ;CFE:
22 CC755 4624730000152020          IZQ   OR,0,6501                          ;
23 CC756 0631000002CC0031          CALL  F,F,ESCAP                          ; CALL ESCAPE(6501)
24                          CF:                                             ;CF:
25 CC757 4662730032CC0400          HZB   OR,EXT,W,WRK0                      ; WRK0:= 12 EXT W(12) CON W(12:23)
26 CC760 0610550000CC0300          CAQ   CAND,B12,W                         ; Q:= W(0:11) CON 12 EXT 0
27 CC761 0607430073CC0000          ZAB   OR,WPRE,WRK1                       ; WRK1:= WPRE
28 CC762 4607101332CC0000          ABB   ADD,SB,WRK0                        ; WRK0:=WRK0+SB
29 CC763 0631000000411004          JMP   T,NEG,CFZER                        ; IF WRK0<0 THEN GOTO CF-ZERO
30 CC764 0653513532CC0400          CAB   SUNO,.23,WRK0,WRK0                 ; WRK0:= WRK0-23
31 CC765 0631000000410773          JMP   T,NEG,CFSHI                        ; IF WRK0<0 THEN GOTO CF-SHIFT
32 CC766 4631000000020775          JMP   F,NZ,CFEXI                         ; IF WRK0=0 THEN GOTO CF-EXIT
33 CC767 0605330001CC0000          ZB    OR,WPRE                            ; IF WPRE = 0 THEN
34 CC770 4631000000021004          JMP   F,NZ,CFZER                         ;   GOTO CF-ZERO
```

```
10032 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 00771 0664713000000000 CFOFL: IZQ   SUNO,4000,0                          ; ELSE SET OVERFLOW
03 00772 7406731224644000        EZB   OR,STOC,STAT,STAT     NEXT TOVIN    ;  EX:= 1,0 , TEST OVERFLOW
04                                                                          ;  TEST INT, GOTO NEXT INSTR
05                               CFSHI:                                     ;CF-SHIFT:
06 00773 0205403500000004        ZAR   ADDO,WRK0,LCD         PUSH          ; LCD:= WRK0 + 1
07 00774 2634330033112060        DRZBB OR,WRK1,F0,NL,F,LCN   LRTN          ;   WRK1,Q:= WRK1(0) CON WRK1,Q(0:22)
08 00775 0605230000000000 CFEXI: ZQ    OR                                 ; IF Q>=0 THEN
09 00776 0631000000411001        JMP   T,NEG,CF1                          ;   EX(22:23):= 0,0 , TEST INT
10 00777 4607431540000000        ZAB   OR,WRK1,W                          ;   W:= WRK1
11 01000 7406731224604200        EZB   OR,STZZ,STAT,STAT     NEXT TSTIN    ;   GOTO NEXT INSTR
12 01001 0607403540000000 CF1:   ZAB   ADDO,WRK1,W                        ; W:= WRK1+1
13 01002 0631000000430771        JMP   T,OVFL,CFOFL                       ; IF OVERFLOW THEN
14 01003 7406731224604200        EZB   OR,STZZ,STAT,STAT     NEXT TSTIN    ; BEGIN
15                                                                          ;   EX(22:23):=0,0 , TEST INT
16                                                                          ;   GOTO NEXT INSTR
17                               CFZER:                                     ;CF-ZERO:
18 01004 0606731224004200        EZB   OR,STZZ,STAT,STAT                  ; EX:= 0
19 01005 3407340000600000        ZBB   AND,W                 NEXT TSTIN    ; W:= 0 , TEST INT, GOTO NEXT INSTR
20
21                               ;MULTIPLY FLOATING, FM, OPCODE = 62
22
23                               FME:                                       ;FME:
24 01006 4624730000144240        IZQ   OR,0,6212                          ;
25 01007 0631000002000031        CALL  F,F,ESCAP                          ; CALL ESCAPE(6212)
26                               FM:                                        ;FM:
27 01010 0662730036000400        HZB   OR,EXT,W,WRK4                      ; WRK4:= 12 EXT W(12) CON W(12:23)
28 01011 0631000000131055        JMP   F,NWADR,FMREG                      ; IF W-ADDR THEN GOTO FM-REG
29 01012 4657730033514000        XZB   OR,DATI,WRK1          CONT TSTIO 1 ; WRK1:= DATAIN , TEST IO
30 01013 0647113426200011        ABBR  SUNO,,2,SB,LAD        CONT READP   ; LAD:=SB:= SB-2 ,  READP
31 01014 0631000000130032        JMP   F,NWADR,INTO                       ; IF W-ADDR THEN GOTO INTO
32 01015 4662731574000400        HZB   OR,EXT,WRK1,WRK2                   ; WRK2:= 12 EXT WRK1(12) CON WRK1(12:23)
33 01016 0613551573000300        CAB   CAND,B12,WRK1,WRK1                 ; WRK1:=WRK1(0:11) CON 12 EXT 0
34 01017 4607103734000000        ABB   ADDO,WRK4,WRK2                     ; WRK2:= WRK2+WRK4+1
35 01020 0657730032514000        XZB   OR,DATI,WRK0          CONT TSTIO 1 ; WRK0:= DATAIN  , TEST IO
```

```
10033 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 01021 0613550000000300 FM1:    CAB    CAND,B12,W,W                          ; W:= W(0:11) CON 12 EXT 0
03 01022 0662530000000600        HAB    OR,LSWAP,W,W                          ; W:= W(0:11) CON W(0:11)
04 01023 4604430000000000        ZAQ    OR,W                                  ; Q:= W
05 01024 0635330001000060        SRZBB  OR,WPRE,Z,NL,F,F                      ; WPRE:= WPRE SHIFT (-1)
06 01025 0635330000001060        SRZBB  OR,W,LNK,NL,F,F                       ; W:= SHLINK CON W(0:22)
07 01026 4662730000000600        HZB    OR,LSWAP,W,W                          ; W:= 12 EXT 0 CON W(0:11)
08 01027 0607430076000000        ZAB    OR,WPRE,WRK4                          ; WRK4:= WPRE
09 01030 4674112041000040        DRABB  SUNO,WPRE,WPRE,Z,MC,F,F               ; MC:= Q(23) C. W(12)  WPRE:= 0
10 01031 4604430000000000        ZAQ    OR,W                                  ; Q:= W
11 01032 4251712000001204        CZR    SUNO,.11,LCD          PUSH            ; FOR LCD:= -11,LCD+1 WHILE LCD<=0 DO
12 01033 0674201540004260        MULQ   ADD,WRK1,NL,F,F                       ; BEGIN
13 01034 2674105501113240        MULD   ADDC,WRK0,WPRE,SGN,MC,F,LCN LRTN      ;   IF ADC=0 THEN
14                                                                            ;   BEGIN
15                                                                            ;     Q:= Q+WRK1
16                                                                            ;     WPRE:=WPRE+WRK0+C
17                                                                            ;   END
18                                                                            ;   ADC:= Q(23)
19                                                                            ;   WPRE,Q:= W(0) CON W,Q(0:22)
20                                                                            ; END
21 01035 0610650000000300        CQQ    CAND,B12                              ; Q:= Q(0:11) CON 12 EXT 0
22 01036 4613541735000300        CAB    AND,B12,WRK4,WRK3                     ; WRK3:= 12 EXT 0 CON WRK4(12:23)
23 01037 0604031640000000        AQQ    OR,WRK3                               ; Q:= Q OR WRK3
24 01040 4251712000001204        CZR    SUNO,.11,LCD          PUSH            ; FOR LCD:=-11,LCD+1 WHILE LCD<= 0 DO
25 01041 0674201540004260        MULQ   ADD,WRK1,NL,F,F                       ;   'MULTIPLY'
26 01042 2674105501113240        MULD   ADDC,WRK0,WPRE,SGN,MC,F,LCN LRTN      ;
27 01043 0610650000000300        CQQ    CAND,B12                              ; Q:= Q(0:11) CON 12 EXT 0
28
29 01044 4662731736000600        HZB    OR,LSWAP,WRK4,WRK4                    ; WRK4:= 12 EXT 0 CON WRK4(0:11)
30 01045 0604031700000000        AQQ    OR,WRK4                               ; Q:= Q OR WRK4
31 01046 0251712000001304        CZR    SUNO,.10,LCD          PUSH            ; FOR LCD:= -10,LCD+1 WHILE LCD<=0 DO
32 01047 0674201540004260        MULQ   ADD,WRK1,NL,F,F                       ;   'MULTIPLY'
33 01050 2674105501113240        MULD   ADDC,WRK0,WPRE,SGN,MC,F,LCN LRTN      ;
```

```
10034 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 01051 0674213540004260          MULQ   SUN0,WRK1,NL,F,F                    ; IF ADC THEN
03 01052 4674115501003260          MULD   SUNC,WRK0,WPRE,SGN,NL,F,F           ; BEGIN
04                                                                            ;   Q:= Q-WRK1
05                                                                            ;   WPRE:= WPRE-WRK0-1+C
06                                                                            ; END
07 01053 0605423600000004          ZAR    SUB0,WRK2,LCD                       ; LCD:= -WRK2 , -EXPONENT
08 01054 0631000000000715          JMP    F,F,NORMA                           ; GOTO NORMALIZE
09
10                                 FMREG:                                     ;FM-REG:
11 01055 4657430173000000          XZAB   CDATI,OR,LAX,WRK1      CONT 0 1     ; WRK1:= LAX
12 01056 4607113426000001          ABBR   SUN0,,2,SB,LAST                     ; SB:=LAST:= SB-2
13 01057 4662731574000400          HZB    OR,EXT,WRK1,WRK2                    ; WRK2:= 12 EXT WRK1(12) CON WRK1(12:23)
14 01060 0613551573000300          CAB    CAND,B12,WRK1,WRK1                  ; WRK1:= WRK1(0:11) CON 12 EXT 0
15 01061 4607103734000000          ABB    ADD0,WRK4,WRK2                      ; WRK2:= WRK2+WRK4+1
16 01062 0607430172000000          ZAB    OR,LAX,WRK0                         ; WRK0:= LAX
17 01063 4631000000001021          JMP    F,F,FM1                             ; GOTO FM1
18
19                                 ;MULTIPLY FLOATING, WHEN THE FPU801 IS PRESENT
20
21                                 FMFPE:                                     ;FMFPE:
22 01064 4624730000144240          IZQ    OR,0,6212                           ;
23 01065 0631000002000031          CALL   F,F,ESCAP                           ; CALL ESCAPE(6212)
24                                 FMFPU:                                     ;FMFPU:
25 01066 4605430000020071          ZAR    OR,W,FPOP1             CONT FMOP   ; FPUOP1:= W   , FMOP
26 01067 0605430040000070          ZAR    OR,WPRE,FPOPU                      ; FPUOPU:= WPRE  , START FPU
27 01070 4631000000131114          JMP    F,NWADR,FPURG                       ; IF W-ADDR THEN GOTO FPU-REG
28 01071 0657730033004000 FPU1:    XZB    OR,DATI,WRK1           CONT 0 1    ; WRK1:= DATAIN  , TEST IO
29 01072 4617730032004100          XZB    OR,IOSTA,WRK0                      ; WRK0:= IOSTA
30 01073 4645113426200011          ABR    SUN0,,2,SB,LAD         CONT READP  ; LAD:= SB-2  , READP
31 01074 0605431540000071          ZAR    OR,WRK1,FPOP1                      ; FPUOP1:= WRK1
32 01075 4631000000131117          JMP    F,NWADR,FPRG1                       ; IF W-ADDR THE GOTO FPRG1
33 01076 0655730000004070          XZR    OR,DATI,FPUPU          CONT 0 1    ; FPOPU:= DATAIN  , TEST IO , START FPU
34 01077 4613730035001400          CZB    OR,,3,WRK3                         ; WRK3:= 3
```

```
I0035 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 01100 0611541500003400          CA     AND,.7,WRK0                          ; IF WRK0(21:23)<> 0 THEN
03 01101 0631000000421424          JMP    T,NZ,INTFP                           ;   GOTO INT_FPU
04 01102 0607113426000000          ABB    SUN0,.2,SB                           ; SB:= SB-2
05 01103 4617730032004100          XZB    OR,IOSTA,WRK0                        ; WRK0:= IOSTA
06 01104 0611541500003400          CA     AND,.7,WRK0                          ; IF WRK0(21:23)<>0 THEN
07 01105 0631000000421424          JMP    T,NZ,INTFP                           ;   GOTO INT_FPU
08 01106 0615730000007000          XZ     OR,FPUR0                             ; MAKE FPUR0 BUSY
09 01107 4657730001007000          XZB    OR,FPUR0,WPRE       CONT 0 1         ; WPRE:= FPUR0
10 01110 4657730000007100          XZB    OR,FPUR1,W          CONT 0 1         ; W:= FPUR1
11 01111 4654541640007200          XAQ    AND,FPUST,WRK3      CONT 0 1         ; Q:= FPUST AND 3
12 01112 0631000000421127          JMP    T,NZ,FPUER                           ; IF Q<>0 THEN GOTO FPU-ERROR
13                                                                             ; EX:= 0
14 01113 7406731224604200          EZB    OR,STZZ,STAT,STAT   NEXT TSTIN       ; TEST INT,  GOTO NEXT INSTR
15
16                           FPURG:                                            ;FPU-REG:
17 01114 0655230000000000          XXZQ   CDATI,OR            CONT 0 1         ; NO OP
18 01115 0605430140000071          ZAR    OR,LAX,FPOP1                         ; FPUOP1:= LAX
19 01116 4607113426000001          ABBR   SUN0,.2,SB,LAST                      ; LAST:=SB:= SB-2
20 01117 4605430140000070 FPRG1:   ZAR    OR,LAX,FPOP0                         ; FPUOP0:= LAX  , START FPU
21 01120 4613730035001400          CZB    OR,.3,WRK3                           ; WRK3:= 3
22 01121 0615730000007000          XZ     OR,FPUR0                             ; MAKE FPUR0 BUSY
23 01122 4657730001007000          XZB    OR,FPUR0,WPRE       CONT 0 1         ; WPRE:= FPUR0
24 01123 4657730000007100          XZB    OR,FPUR1,W          CONT 0 1         ; W:= FPUR1
25 01124 4654541640007200          XAQ    AND,FPUST,WRK3      CONT 0 1         ; Q:= FPUST AND 3
26 01125 0631000000421127          JMP    T,NZ,FPUER                           ; IF Q<>0 THEN GOTO FPU-ERROR
27                                                                             ; EX:= 0
28 01126 7406731224604200          EZB    OR,STZZ,STAT,STAT   NEXT TSTIN       ; TEST INT , GOTO NEXT INSTR
29
30                           FPUER:                                            ;FPU-ERROR:
31 01127 4606631224004200          EQB    OR,STZZ,STAT,STAT                    ; STAT(21:23):= Q
32 01130 4611541200002600          CA     AND,.1B5,STAT                        ; IF STAT(5) THEN
33 01131 0631000000420035          JMP    T,NZ,INTF                            ;   GOTO FLOATING EXCEPTION
34 01132 7405230000600000          ZQ     OR                  NEXT TSTIN       ; ELSE TEST INT , GOTO NEXT INSTR
```

```
1003e CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;DIVIDE FLOATING, FD, OPCODE = 64
03
04                          FDE:                                                    ;FDE:
05 01133 4624730000150240           IZQ    OR,0,6412                            ;
06 01134 0631000002CC0031           CALL   F,F,ESCAP                            ; CALL ESCAPE(6412)
07                          FD:                                                     ;FD:
08 01135 0662730035CC0400           HZB    OR,EXT,W,WRK3                        ; WRK3:= 12 EXT W(12) CON W(12:23)
09 01136 4631000000131200           JMP    F,NWADR,FDREG                        ; IF W=ADDR THEN GOTO FD-REG
10 01137 4657730033514000           XZB    OR,DATI,WRK1          CONT TSTIO 1   ; WRK1:= DATAIN  , TEST IO
11 01140 0647113426200011           ABBR   SUNO,,2,SB,LAD        CONT READP     ; LAD:= SB:= SB-2  , READP
12 01141 0631000000130032           JMP    F,NWADR,INTU                         ; IF W=ADDR THEN GOTO INTU
13 01142 4662731574CC0400           HZB    OR,EXT,WRK1,WRK2                     ; WRK2:= 12 EXT WRK1(12) CON WRK1(12:23)
14 01143 0613551573CC0300           CAB    CAND,B12,WRK1,WRK1                   ; WRK1:= WRK1(0:11) CON 12 EXT 0
15 01144 0657730032514000           XZB    OR,DATI,WRK0          CONT TSTIO 1   ; WRK0:= DATAIN  , TEST IO
16 01145 4631000000021206 FDO:      JMP    F,NZ,FDZER                           ; IF WRK0=0 THEN GOTO FD-ZERO_DIVISOR
17 01146 4605430040CC0000           ZA     OR,WPRE                              ; IF WPRE = 0 THEN
18 01147 0631000000020721           JMP    F,NZ,CIZER                           ;   GOTO CI-ZERO
19 01150 0613550000CC0300           CAB    CAND,B12,W,W                         ; W:= W(0:11) CON 12 EXT 0
20 01151 0607123674000000           ABB    SUBO,WRK3,WRK2                       ; WRK2:= WRK3 - WRK2
21 01152 0637330032CC4000           NSZBB  OR,WRK0,DVS,F,F                      ; DVS:= WRK0(0)
22 01153 4637330001CC4020           NSZBB  OR,WPRE,DC,F,F                       ; ADC:= WPRE(0) EXOR -,DIVSIGN
23 01154 0604240000CC0000           ZQQ    AND                                  ; Q:= 0
24 01155 0251712000CC0404           CZR    SUNO,,23,LCD          PUSH           ; FOR LCD:=-23,LCD+1 WHILE LCD<=0 DO
25 01156 4676107540002160           DIVD   ADDX,WRK1,W,ADC,NL,F,F               ; BEGIN C. DIVIDE
26 01157 6677105501111120           DIVS   ADDC,WRK0,WPRE,LNK,DC,F,LCN LRTN     ;   IF ADC = 1 THEN
27                                                                              ;   BEGIN
28                                                                              ;     W:= W-WRK1
29                                                                              ;     WPRE:= WPRE-WRK0-1+C
30                                                                              ;   END ELSE
31                                                                              ;   BEGIN
32                                                                              ;     W:= W+WRK1
33                                                                              ;     WPRE:= WPRE+WRK0
34                                                                              ;   END
35                                                                              ;   SHLNK CON WPRE CON W CON Q:=
36                                                                              ;   WPRE CON W CON Q CON 1 EXT ADC
37                                                                              ;   ADC:= WPRE(0) EXOR -,DIVSIGN
38                                                                              ; END
```

```
10037 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 01160 0613660035CC3500          CQB   EXOR,.1B0,WRK3                    ; WRK3:= Q EXOR BIT 0
03 01161 4631000000461166          JMP   T,NORMU,FD1                       ; IF WRK3 NOT NORMALIZED THEN
04 01162 4637330035CC2060          SLZBB OR,WRK3,ADC,NL,F,F                ; BEGIN
05 01163 4677107540CC0160          DIVS  ADDX,WRK1,W,Z,NL,F,F              ;   SH LINK CON WRK3:= WRK3 CON ADC
06 01164 0677105501CC1120          DIVS  ADDC,WRK0,WPRE,LNK,DC,F,F         ;   ' DIVIDE '
07 01165 0607310034CC0000          ZBB   SUN,WRK2                          ;   WRK2:= WRK2-1
08                            FD1:                                         ; END
09 01166 4610730000CC0300          CZQ   OR,B12                            ; Q:= 8'7777
10 01167 4251712000001204          CZR   SUN0,.11,LCD          PUSH        ; FOR LCD:=-11,LCD+1!WHILE LCD<=0 DO
11 01170 4676107540CC2160          DIVD  ADDX,WRK1,W,ADC,NL,F,F            ;   ' DIVIDE '
12 01171 6677105501111120          DIVS  ADDC,WRK0,WPRE,LNK,DC,F,LCN LRTN  ;
13 01172 4607230000CC0000          ZQB   OR,W                              ; W:= Q
14 01173 4662730000CC0500          HZB   OR,SWAP,W,W                       ; W:= W(12:23) CON W(0:11)
15 01174 4605421600CC0004          ZAR   SUB,WRK2,LCD                      ; LCD:= -WRK2 -1
16 01175 0607431641CC0000          ZAB   OR,WRK3,WPRE                      ; WPRE:= WRK3
17 01176 4604406000CC0000          ZAQ   ADDX,W                            ; Q:= W + ADC
18 01177 4631000000C00730          JMP   F,F,ROUND                         ; GOTO ROUND
19
20                            FDREG:                                       ;FD-REG:
21 01200 4657430173CC0000          XZAB  CDATI,OR,LAX,WRK1     CONT 0 1    ; WRK1:= LAX
22 01201 4607113426CC0001          ABBR  SUN0,.2,SB,LAST                   ; SB:=LAST:= SB-2
23 01202 4662731574CC0400          HZB   OR,EXT,WRK1,WRK2                  ; WRK2:= 12 EXT WRK1(12) CON WRK1(12:23)
24 01203 0613551573000300          CAB   CAND,B12,WRK1,WRK1                ; WRK1:= WRK1(0:11) CON 12 EXT 12
25 01204 0607430172000000          ZAB   OR,LAX,WRK0                       ; WRK0:= LAX
26 01205 4631000000001145          JMP   F,F,FD0                           ; GOTO FD0
27
28                            FDZER:                                       ;FD-ZERO_DIVISOR:
29 01206 0664713000000000          IZQ   SUN0,4000,0                       ; SET OVERFLOW
30 01207 4606731224CC4000          EZB   OR,STOC,STAT,STAT                 ; EX(22:23):= 2
31 01210 0631000000C00752          JMP   F,F,TINTF                         ; GOTO TEST-INTF
```

```
10038 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;FLOATING DIVIDE, FD, WHEN THE FPU801 IS PRESENT
03
04                          FDFPE:                                              ;FDFPE:
05 01211 4624730000150240           IZQ    OR,0,6412                        ;
06 01212 0631000002000031           CALL   F,F,ESCAP                        ; CALL ESCAPE(6412)
07                          FDFPU:                                              ;FDFPU:
08 01213 0605430000030071           ZAR    OR,W,FPOP1          CONT FDOP    ; FPUOP1:= W  , FDOP
09 01214 0605430040000070           ZAR    OR,WPRE,FPOPU                    ; FPUOPO:= WPRE  , START FPU
10 01215 4631000000131114           JMP    F,NWADR,FPURG                    ; IF W-ADDR THEN GOTO FPU-REG
11 01216 4631000000001071           JMP    F,F,FPU1                         ; GOTO FPU1
12
13                          ;FLOATING ADD, FA, OPCODE = 60
14
15                          FAE:                                                ;FAE:
16 01217 0624730000140240           IZQ    OR,0,6012                        ;
17 01220 0631000002000031           CALL   F,F,ESCAP                        ; CALL ESCAPE(6012)
18                          FA:                                                 ;FA:
19 01221 0662730035000400           HZB    OR,EXT,W,WRK3                    ; WRK3:= 12 EXT W(12) CON W(12:23)
20 01222 4631000000131256           JMP    F,NWADR,FAREG                    ; IF W-ADDR THEN GOTO FA-REG
21 01223 4657730033514000           XZB    OR,DATI,WRK1        CONT TSTIO 1 ; WRK1:= DATAIN  , TEST IO
22 01224 0647113426200011           ABBR   SUNO,,2,SB,LAD      CONT READP   ; LAD:=SB:= SB-2  , READP
23 01225 0631000000130032           JMP    F,NWADR,INTO                     ; IF W-ADDR THEN GOTO INTO
24 01226 4662731574000400           HZB    OR,EXT,WRK1,WRK2                 ; WRK2:= 12 EXT WRK1(12) CON WRK1(12:23)
25 01227 0613551573000300           CAB    CAND,B12,WRK1,WRK1               ; WRK1:= WRK1(0:11) CON 12 EXT 0
26 01230 0657730032514000           XZB    OR,DATI,WRKO        CONT TSTIO 1 ; WRKO:= DATAIN  , TEST IO
27 01231 0613550000000300 FA1:      CAB    CAND,B12,W,W                     ; W:= W(0:11) CON 12 EXT 0
28 01232 0607113635000000           ABB    SUNO,WRK2,WRK3                   ; IF WRK3>WRK2 THEN
29 01233 0631000000021251           JMP    F,NZ,FAEQ                        ; BEGIN
30 01234 4631000000411245           JMP    T,NEG,FASM                       ;
31 01235 0607101674000000           ABB    ADD,WRK3,WRK2                    ;   WRK2:= WRK3
32 01236 4651513640000200           CA     SUNO,,38,WRK3                    ;   IF WRK3>= 38 THEN
33 01237 0631000000011264           JMP    F,NEG,FSAEX                      ;     GOTO EXIT
34 01240 4604431540000000           ZAQ    OR,WRK1                          ;   Q:= WRK1
35 01241 4251523640003704           CAR    SUBO,,1,WRK3,LCD    PUSH         ;   LCD:= -(WRK3-WRK2) +1
36 01242 2634330032522060           DRZBB  OR,WRKO,FO,NL,1,MAXLO LRTN        ;   WRKO,Q:= SGN EXT WRKO,Q(0:23-COUNT)
37 01243 4607230033000000           ZQB    OR,WRK1                          ;   WRK1:= Q
38 01244 4631000000001251           JMP    F,F,FAEQ                         ; END ELSE
```

```
10039 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 01245 4604430000000000 FASM:  ZAQ   OR,W                                  ; IF WRK2>WRK3 THEN
03 01246 4205403640000004        ZAR   ADDO,WRK3,LCD          PUSH          ; BEGIN
04 01247 2634330001522060        DRZBB OR,WPRE,FO,NL,T,MAXLO LRTN           ;   Q:= W
05 01250 4607230000000000        ZQB   OR,W                                 ;   LCD:= WRK3-WRK2+1
06                                                                          ;   COUNT:= MIN(-LCD,48)
07                                                                          ;   WPRE,Q:= SGN EXT WPRE,Q(0:23+COUNT)
08                                                                          ;   W:= Q
09                               FAEQ:                                      ; END
10 01251 4604430000000000        ZAQ   OR,W                                 ; Q:= W
11 01252 0604001540000000        AQQ   ADD,WRK1                             ; Q:= Q+WRK1
12 01253 4674105501003060        DRABB ADDC,WRK0,WPRE,SGN,NL,F,F            ; WPRE,Q:= ((WPRE+WRK0+C),Q) SHIFT (-1)
13 01254 4605421600000004        ZAR   SUB,WRK2,LCD                         ; LCD:= -WRK2 - 1
14 01255 0631000000000715        JMP   F,F,NORMA                            ; GOTO NORMALIZE
15
16                               FAREG:                                     ;FA-REG:
17 01256 4657430173000000        XZAB  CDATI,OR,LAX,WRK1      CONT U 1      ; WRK1:= LAX
18 01257 4607113426000001        ABBR  SUNO,,2,SB,LAST                      ; SB:=LAST:= SB-2
19 01260 4662731574000400        HZB   OR,EXT,WRK1,WRK2                     ; WRK2:= 12 EXT WRK1(12) CON WRK1(12:23)
20 01261 0613551573000300        CAB   CAND,B12,WRK1,WRK1                   ; WRK1:= WRK1(0:11) CON 12 EXT 0
21 01262 0607430172000000        ZAB   OR,LAX,WRK0                          ; WRK0:= LAX
22 01263 4631000000001231        JMP   F,F,FA1                              ; GOTO FA1
23                               FSAEX:                                     ;EXIT:
24 01264 4613541634000300        CAB   AND,B12,WRK2,WRK2                    ; WRK2:= WRK2(12:23)
25 01265 7407131600600000        ABB   OR,WRK2,W              NEXT TSTIN    ; W:= W OR WRK2 ,  TEST INT
26                                                                          ; GOTO NEXT INST
27
28                               ;FLOATING ADD, FA, WHEN THE FPU801 IS PRESENT
29
30                               FAFPE:                                     ;FAFPE:
31 01266 0624730000140240        IZQ   OR,0,6012                            ;
32 01267 0631000002000031        CALL  F,F,ESCAP                            ; CALL ESCAPE(6012)
33                               FAFPU:                                     ;FAFPU:
34 01270 0605430000000071        ZAR   OR,W,FPOP1             CONT FAOP     ; FPUOP1:= W  , FAOP
35 01271 0605430040000070        ZAR   OR,WPRE,FPOPU                        ; FPUOP0:= WPRE  , START FPU
36 01272 4631000000131114        JMP   F,NWADR,FPURG                        ; IF W=ADDR THEN GOTO FPU-REG
37 01273 4631000000001071        JMP   F,F,FPU1                             ; GOTO FPU1
```

```
1004C CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;FLOATING SUBTRACT, FS, OPCODE = 61
03
04                          FSE:                                                ;FSE:
05 01274 4624730000142240           IZQ   OR,0,6112                           ;
06 01275 0631000002CC0031           CALL  F,F,ESCAP                           ; CALL ESCAPE(6112)
07                          FS:                                                 ;FS:
08 01276 0662730035CC0400           HZB   OR,EXT,W,WRK3                       ; WRK3:= 12 EXT W(12) CON W(12:23)
09 01277 0631000000131333           JMP   F,NWADR,FSREG                       ; IF W-ADDR THEN GOTO FS-REG
10 01300 4657730033514000           XZB   OR,DATI,WRK1          CONT TSTIO 1  ; WRK1:= DATAIN  , TEST IO
11 01301 0647113426200011           ABBR  SUNO,,2,SB,LAD        CONT READP    ; LAD:=SB:= SB-2  , READP
12 01302 0631000000130032           JMP   F,NWADR,INTU                        ; IF W-ADDR THEN GOTO INTU
13 01303 4662731574CC0400           HZB   OR,EXT,WRK1,WRK2                    ; WRK2:= 12 EXT WRK1(12) CON WRK1(12:23)
14 01304 0613551573CC0300           CAB   CAND,B12,WRK1,WRK1                  ; WRK1:= WRK1(0:11) CON 12 EXT 0
15 01305 0657730032514000           XZB   OR,DATI,WRK0          CONT TSTIO 1  ; WRK0:= DATAIN  , TEST IO
16 01306 0613550000CC0300 FS1:      CAB   CAND,B12,W,W                        ; W:= W(0:11) CON 12 EXT 0
17 01307 0607113635CC0000           ABB   SUNO,WRK2,WRK3                      ;
18 01310 4631000000021326           JMP   F,NZ,FSEQ                           ; IF WRK3>WRK2 THEN
19 01311 4631000000411322           JMP   T,NEG,FSSM                          ; BEGIN
20 01312 0607101674CC0000           ABB   ADD,WRK3,WRK2                       ;   WRK2:= WRK3
21 01313 4651513640CC0200           CA    SUNO,,38,WRK3                       ;   IF WRK3>= 38 THEN
22 01314 0631000000011264           JMP   F,NEG,FSAEX                         ;     GOTO EXIT
23 01315 4604431540CC0000           ZAQ   OR,WRK1                             ;   Q:= WRK1
24                                                                            ;   COUNT:= MIN(48,WRK3-WRK2)
25 01316 4251523640CC3704           CAR   SUBO,,1,WRK3,LCD      PUSH          ;   LCD:= -(WRK3-WRK2)+1
26 01317 2634330032522060           DRZBB OR,WRK0,FO,NL,T,MAXLO LRTN          ;   WRK0,Q:= SGN EXT WRK0,Q(0:23-COUNT)
27 01320 4607230033CC0000           ZQB   OR,WRK1                             ;   WRK1:= Q
28 01321 0631000000CC1326           JMP   F,F,FSEQ                            ; END ELSE
29 01322 4604430000CC0000 FSSM:     ZAQ   OR,W                                ; IF WRK2>WRK3 THEN
30 01323 4205403640CC0004           ZAR   ADDO,WRK3,LCD         PUSH          ; BEGIN
31 01324 2634330001522060           DRZBB OR,WPRE,FO,NL,T,MAXLO LRTN          ;   Q:= W
32                                                                            ;   COUNT:=MIN(48,WRK2-WRK3)
33                                                                            ;   LCD:=WRK3-WRK2+1
34                                                                            ;   WPRE,Q:= SGN EXT WPRE,Q(0:23-COUNT)
35 01325 4607230000CC0000           ZQB   OR,W                                ;   W:= Q
36                                                                            ; END
```

```

                         FSEQ:                                                      ;
01326 46044300000000000         ZAQ   OR,W                                          ; Q:= W
01327 06040135400000000         AQQ   SUNO,WRK1                                     ; Q:= Q - WRK1
01330 06741155010003060         DRABB SUNC,WRKO,WPRE,SGN,NL,F,F                     ; WRKO,Q:= (WPRE - WRKO-1+C,Q) SHIFT (-1)
01331 46054216000000004         ZAR   SUB,WRK2,LCD                                  ; LCD:= -WRK2-1
01332 06310000000000715         JMP   F,F,NORMA                                     ; GOTO NORMALIZE
                         FSREG:                                                     ;FS-REG:
01333 46574301730000000         XZAB  CDATI,OR,LAX,WRK1       CONT 0 1              ; WRK1:= LAX
01334 46071134260000001         ABBR  SUNO,.2,SB,LAST                               ; SB:= LAST:= SB-2
01335 46627315740000400         HZB   OR,EXT,WRK1,WRK2                              ; WRK2:= 12 EXT WRK1(12) CON WRK1(12:23)
01336 06135515730000300         CAB   CAND,B12,WRK1,WRK1                            ; WRK1:= WRK1(0:11) CON 12 EXT 0
01337 06074301720000000         ZAB   OR,LAX,WRKO                                   ; WRKO:= LAX
01340 46310000000001306         JMP   F,F,FS1                                       ; GOTO FS1

                         ;FLOATING SUBTRACT, FS, WHEN THE FPU801 IS PRESENT

                         FSFPE:                                                     ;FSFPE:
01341 46247300000142240         IZQ   OR,0,6112                                     ;
01342 06310000020000031         CALL  F,F,ESCAP                                     ; CALL ESCAPE(6112)
                         FSFPU:                                                     ;FSFPU:
01343 46054300000010071         ZAR   OR,W,FPOP1              CONT FSOP             ; FPUOP1:= W   , FSOP
01344 06054300400000070         ZAR   OR,WPRE,FPOPO                                 ; FPUOPO:= WPRE
01345 46310000000131114         JMP   F,NWADR,FPURG                                 ; IF W-ADDR THEN GOTO FPU-REG
01346 46310000000001071         JMP   F,F,FPU1                                      ; GOTO FPU1
```

```
10042 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;RETURN FROM ESCAPE, RE, OPCODE = 26
03
04                          REE:                                                ;REE:
05 01347 0624730000054640           IZQ    OR,0,2632                        ;
06 01350 0631000002000031           CALL   F,F,ESCAP                        ; CALL ESCAPE(2632)
07                          RE:                                                 ;RE:
08 01351 0631000000130032           JMP    F,NWADR,INTU                     ; IF W_ADDR THEN GOTO INTU
09 01352 0657730020514000           XZB    OR,DATI,W0          CONT TSTIO 1 ; W0:= DATI, TSTIO;
10 01353 0647101426200011           ABBR   ADD,.2,SB,LAD       CONT READP   ; LAD,SB:= SB+2, READP;
11 01354 0657730021514000           XZB    OR,DATI,W1          CONT TSTIO 1 ; W1:= DATI, TSTIO;
12 01355 0647101426200011           ABBR   ADD,.2,SB,LAD       CONT READP   ; LAD,SB:= SB+2, READP
13 01356 0657730022514000           XZB    OR,DATI,W2          CONT TSTIO 1 ; W2:= DATI, TSTIO;
14 01357 0647101426200011           ABBR   ADD,.2,SB,LAD       CONT READP   ; LAD,SB:= SB+2, READP;
15 01360 0657730023514000           XZB    OR,DATI,W3          CONT TSTIO 1 ; W3:= DATI, TSTIO;
16 01361 0647101426200011           ABBR   ADD,.2,SB,LAD       CONT READP   ; LAD,SB:= SB+2, READP;
17 01362 0657730024514000           XZB    OR,DATI,STAT        CONT TSTIO 1 ; STAT:= DATI, TSTIO;
18 01363 0647101426200011           ABBR   ADD,.2,SB,LAD       CONT READP   ; LAD,SB:= SB+2, READP;
19 01364 0657730033514000           XZB    OR,DATI,WRK1        CONT TSTIO 1 ; WRK1:= DATI, TSTIO; LATER IC
20 01365 0647101426200011           ABBR   ADD,.2,SB,LAD       CONT READP   ; LAD,SB:= SB+2, READP
21 01366 0657730025514000           XZB    OR,DATI,CAUSE       CONT TSTIO 1 ; CAUSE:= DATI, TSTIO;
22 01367 0647101426200011           ABBR   ADD,.2,SB,LAD       CONT READP   ; LAD,SB:= SB+2, READP;
23 01370 0657730034514000           XZB    OR,DATI,WRK2        CONT TSTIO 1 ; WRK2:= DATI, TSTIO; LATER SB
24 01371 4607431626000000           ZAB    OR,WRK2,SB                       ; SB:= WRK2
25 01372 0605431540240011           ZAR    OR,WRK1,LAD         CONT READJ   ; LAD:= WRK1  , READ INSTRUCTION
26 01373 0631000000561375           JMP    T,MMODE,RGSET                    ; IF NOT MONMODE THEN
27 01374 0613551224003500           CAB    CAND,.1B0,STAT,STAT              ;   CLEAR MONMODE
28 01375 0613551224000005 RGSET:    CABR   CAND,B1214,STAT,STAT,CPUST       ; STAT:=CPUST:= STAT CAND 8'7000
29                                                                           ; C. REMOVE DUMPERRORCOUNT
30 01376 0631000000130032           JMP    F,NWADR,INTU                     ; IF W_ADDR THEN GOTO INTU
31 01377 0605431540000003           ZAR    OR,WRK1,ICD                      ; ICD:= WRK1
32 01400 7455730000610100           XZ     OR,CIOST            NEXT TLIIN 1 ; TEST LIMIT, TEST INT
33                                                                           ; GOTO NEXT INSTR
```

```
10043 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;BUS INTERRUPTS
03                          ;***************
04
05                          ;PFINT:                                                 ;PF-INT:  C. PREFETCH IMTERRUPT
06                          .FINT:
07 01401 461373002500 1300          CZB   OR,.10,CAUSE                            ; CAUSE:= 10
08 01402 461773003200 4100          XZB   OR,IOSTA,WRK0                           ; WRK0:= IOSTATUS
09 01403 462473000000 0000          IZQ   OR,0,3400                               ;
10 01404 460704153200 0000          AQB   AND,WRK0,WRK0                           ; WRK0:= WRK0(13:15) C. PF-STATUS
11 01405 025171200000 3404          CZR   SUN0,.7,LCD          PUSH               ;
12 01406 663533003211 0060          SRZBB OR,WRK0,Z,NL,F,LCN   LRTN               ; WRK0:= 21 EXT 0 CON WRK0(13:15)
13 01407 462473177760 5200          IZQ   OR,7777,250                             ;
14 01410 460704122400 0000          AQB   AND,STAT,STAT                           ; CLEAR EX BUT BIT 16,18,20
15 01411 060713152400 0000          ABB   OR,WRK0,STAT                            ; STAT:= STAT OR WRK0
16 01412 463100000000 1431          JMP   F,F,TESTI                               ; GOTO TEST-INT
17
18                          INTB2:                                                ;INTB2:  C. ERROR IN REGISTER DUMP
19 01413 465350362600 3700          CAB   ADD0,.1,WRK2,SB                         ; SB:= WRK2+2  C. DUMPADDRESS
20 01414 463100000000 1423          JMP   F,F,INTB                                ;
21                          INTBB:                                                ;INTBB: C. BUS INTERRUPT FROM TEST
22 01415 461154120000 3200          CA    AND,.1B2,STAT                           ; IF AFTER AM=1 THEN
23 01416 063100000002 1423          JMP   F,NZ,INTB                               ;   IC:= IC+2
24 01417 464550140000 1003          RAR   ADD,ICS,.2,ICD                          ;   C. ERROR IN INDIRECT ADDRESS CALC
25 01420 463100000000 1423          JMP   F,F,INTB                                ; GOTO INTB
26                          INTB3:                                                ;INTB3  C. ERROR IN FETCH OF SERVICE
27                                                                                ;          ADDRESS
28 01421 060743136600 0000          ZAB   OR,INF,SB                               ; SB:= INF
29 01422 465351336700 1100          CAB   SUN0,.12,INF,INF                        ; INF:= INF-12
30                          INTB:                                                 ;INTB:
31 01423 461773003200 4100          XZB   OR,IOSTA,WRK0                           ; WRK0:= IOSTA
```

```
10044 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;INTB1:                                                 ;INTB1:
03                          .NTB1:                                                  ;
04                          INTFP:                                                  ;INTF:  C. ENTRY FROM FPU
05 01424 0613730025CC3600           CZB   OR,.8,CAUSE                             ; CAUSE:= 8
06 01425 4624731777605200           IZQ   OR,7777,250                             ;
07 01426 4607041224CC0000           AQB   AND,STAT,STAT                           ; CLEAR EX BUT BIT 16,18,20
08 01427 0613541532CC3400           CAB   AND,.7,WRKU,WRK0                        ; WRK0:= 21 EXT U CON WRKU(21:23)
09 01430 0607131524CC0000           ABB   OR,WRKU,STAT                            ; STAT:= STAT OR WRKU
10                          TESTI:                                                  ;TEST-INT:
11 01431 0601730000CC0400           SZ    OR,ILIMC                                ; IF ILIMC<>U THEN
12 01432 4631000000421434           JMP   T,NZ,ICALL                              ;   GOTO INTERRUPTCALL
13 01433 4631000000CC1767           JMP   F,F,SYFLO                               ; ELSE GOTO SYSTEMFAULTO
14
15                          ;EXTERNAL INTERRUPTS AND MONITOR CALLS
16                          ;**************************************
17
18                          ICALL:                                                  ;INTERRUPTCALL:
19 01434 0631000002CC1533           CALL  F,F,LOKM                                ; LOCKMONITOR;
20 01435 0607131524CC0000           ABB   OR,WRK0,STAT                            ; STAT:= STAT OR WRK0; UNLOCK
21 01436 0605431340210011           ZAR   OR;INF,LAD              CONT READ       ; LAD:= INF, READ;
22 01437 4600730000CC0515           SZQR  OR,SIZE,ULIM                            ; ULIM:= SIZE
23 01440 0641220000CC0302           ZQS   SUB,ULIMC                               ; ULIMC:= SIZE
24 01441 4610730000CC3614           CZQR  OR,.8,LLIM                              ; LLIM:= 8
25 01442 4641220000CC0202           ZQS   SUB,LLIMC                               ; LLIMC:= 8
26 01443 4604240000CC0013           ZQQR  AND,BASE                                ; BASE:= U
27 01444 4641220000CC0102           ZQS   SUB,BASEC                               ; BASEC:= 0
28 01445 0657730036CC4000           XZB   OR,DATI,WRK4            CONT 0 1        ; WRK4:= DATAIN  C. NEW STATUS
29 01446 0615541740CC4100           XA    AND,IOSTA,WRK5                          ; IF IOSTATUS(21:23)<>0 THEN
30 01447 4631000000420034           JMP   T,NZ,SYFL1                              ;   GOTO SYSTEMFAULT1
31 01450 0611541200CC2100           CA    AND,.1B16,STAT                          ; IF STAT(16) = 0 THEN
32 01451 4631000000421462           JMP   T,NZ,ICAL2                              ; BEGIN <* NOT USER MODE *>
33 01452 4645101427210011           ABR   ADD,.2,INF,LAD          CONT READ       ;  LAD:=INF+2  , READ
34 01453 4653513375CC1400           CAB   SUNO,.3,INF,WRK3                        ;  WRK3:= INF - 3  C. NEW IC
35 01454 4653513367CC1100           CAB   SUNO,.12,INF,INF                        ;  INF:= INF-12
```

```
10045 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 01455 4613531367003700          CAB   OR,.1,INF,INF                          ;  INF:= INF OR 1
03 01456 0717523434004000          XAB   SUB0,DATI,.2,WRK2      CONT 0 2        ;  WRK2:= DATAIN - 2  C. DUMP ADDR
04 01457 0615541740004100          XA    AND,IOSTA,WRK5                         ;  IF IOSTATUS(21:23)<>0 THEN
05 01460 4631000000421524          JMP   T,NZ,DMPFL                             ;    GOTO DUMPFAULT
06 01461 0631000000001471          JMP   F,F,ICAL3                              ; END ELSE
07                           ICAL2:                                             ; BEGIN
08 01462 4640523400001300          SAQ   SUB0,DMOFS,.2                          ;
09 01463 0643600034000700          SQB   ADD,CUR,WRK2                           ;  WRK2:= CUR + DUMP OFFSET - 2;
10 01464 4653513375001400          CAB   SUN0,.3,INF,WRK3                       ;  WRK3:= INF - 3; NEW IC
11 01465 4653513367001100          CAB   SUN0,.12,INF,INF                       ;  INF:= INF - 12;
12 01466 4613531367003700          CAB   OR,.1,INF,INF                          ;  INF:= INF OR 1;
13 01467 0611541440001500          CA    AND,.4,CNTR                            ;  IF CNTR(21)<>0 THEN <*NOT RUN*>
14 01470 0631000000421502          JMP   T,NZ,ICAL5                             ;    SKIP REGISTER DUMP
15                           ICAL3:                                             ; END;
16 01471 0624730000034320          IZQ   OR,0,GRTB/3-1                          ; Q:= ADDR(GRTB-1)
17 01472 0207431032000000          ZAB   OR,WU,WRK0             PUSH            ; FOR I:= 0 TO 8 DO
18                                                                              ; BEGIN
19 01473 4605431500000012          ZAR   OR,WRK0,DATO                           ;   DATAOUT:= WRK0
20 01474 0647101434230011          ABBR  ADD,.2,WRK2,LAD        CONT WRT        ;   WRK2,LAD:= WRK2, WRT;
21 01475 4630202003000000          CXZQQ ADD0,F,F                               ;   WRK0:=GETREG(Q+1), Q:= Q+1
22 01476 0655541740000100          XA    AND,CIOST,WRK5         CONT 0 1        ;   IF CPUIOSTATUS(21:23)<>0 THEN
23 01477 2631000000421524          JMP   T,NZ,DMPFL             LRTN            ;      GOTO DUMPFAULT
24                                                                              ; END
25                           ICAL4:                                             ;
26 01500 0655541740000100          XA    AND,CIOST,WRK5         CONT 0 1        ; IF IOSTATUS(21:23)<>0 THEN
27 01501 4631000000421524          JMP   T,NZ,DMPFL                             ;   GOTO DUMPFAULT
28                           ICAL5:                                             ;
29 01502 0605431640210011          ZAR   OR,WRK3,LAD            CONT READ       ; LAD:= WRK3  , READ
30 01503 4610541700000321          CAQR  AND,B12,WRK4,ILIM                      ; ILIM:= WRK4(12:23)
31 01504 4641220000000402          ZQS   SUB,ILIMC                              ; ILIMC:= WRK4(12:23)
32 01505 0613551733000305          CABR  CAND,B12,WRK4,WRK1,CPUST               ; WRK1:=CPUST:= WRK4(0:11) CON 12 EXT 0
33 01506 4610541200001700          CAQ   AND,.1B18,STAT                         ; Q:= STAT(18); SAVE PROCESS ACTIVE
```

```
10046 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 01507 4655541740004100          XA     AND,IOSTA,WRK5          CONT 0 1      ; IF IOSTATUS(21:23)<>0 THEN
03 01510 4631000000421524          JMP    T,NZ,DMPFL                            ;   GOTO DUMPFAULT
04 01511 0617730032244011          XZBR   OR,DATI,WRK0,LAD        CONT READJ    ; LAD:=WRK0:= DATI  , READ INSTRUCTION
05 01512 0631000000130032          JMP    F,NWADR,INT0                          ; IF W-ADDR THEN GOTO INT0
06 01513 4605431500000003          ZAR    OR,WRK0,ICD                           ; ICD:= WRK0
07 01514 0607431262000000          ZAB    OR,CAUSE,W2                           ; W2:= CAUSE
08 01515 0613541232000000          CAB    AND,B1214,STAT,WRK0                   ; IF DUMPERRORCOUNT <> 0 THEN
09 01516 0631000000021521          JMP    F,NZ,DMPOK                            ;   W2:= 4
10 01517 0607131533000000          ABB    OR,WRK0,WRK1                          ;   WRK1:= WRK1 OR STAT(12.,14);
11 01520 0613730022001500          CZB    OR,.4,W2                              ;
12                           DMPOK:                                            ;DUMPOK:
13 01521 0607031564000000          AQB    OR,WRK1,STAT                          ; STAT:= WRK1 OR Q;
14 01522 4613551471001506          CABR   CAND,.4,CNTR,CNTR,CNTR0               ; CNTR(21),CNTR0(21):= 0;
15                                                                               ; <* SET RUN LAMP AND FLAG *>
16 01523 7407431621000000          ZAB    OR,WRK2,W1              NEXT          ; W1:= WRK2  C. TOP DUMP ADDR
17                                                                               ; GOTO NEXT INSTR
18                           DMPFL:                                            ;DUMPFAULT:
19 01524 4653501224002200          CAB    ADD,.1B14,STAT,STAT                   ; INCREASE DUMPERRORCOUNT
20 01525 4610541700000300          CAQ    AND,B12,WRK4                          ;
21 01526 4641220000000402          ZQS    SUB,ILIMC                             ; ILIMC:= WRK4(12:23) NEW ILIM
22 01527 0613551224002100          CAB    CAND,.1B16,STAT,STAT                  ; STAT(16):= 0; CLEAR USERMODE
23 01530 0611541200002200          CA     AND,.1B14,STAT                        ; IF DUMPERROR OVERRUN THEN <* > 1*>
24 01531 0631000000021773          JMP    F,NZ,SYFL2                            ;   GOTO SYSTEMFAULT2
25 01532 4631000000001431          JMP    F,F,TESTI                             ; ELSE GOTO TEST-INT
```

```
10047 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          LOKM:                                          ;LOCKMONITOR:
03 01533 4613730037CC3400           CZB   OR,.7,WRK5                      ; WRK5:= 7
04                          LOKM1:                                         ;
05 01534 0611730000CC1424           CZR   OR,.3,CAMBC                      ; SET BYPASSCAM AND BUSBUSY
06 01535 4653513375213011           CABR  SUN0,.6,INF,WRK3,LAD  CONT READ  ; LAD,WRK3:= INF-6, READ;
07 01536 4607240032CC0000           ZQB   AND,WRK0                         ; CLEAR NOT-UNLOCK CONDITION;
08 01537 0601730000CC0712           SZR   OR,CUR,DATO                      ; DATO:= CUR;
09 01540 0654730000504000           XZQ   OR,DATI               CONT TSTBE 1 ; Q:= DATI, TEST BUSERROR;
10 01541 4631000000421547           JMP   T,NZ,LOKM2                       ; IF Q <> 0 THEN GOTO LOKM2;
11 01542 4611730000003724           CZR   OR,.1,CAMBC                      ; CAMBC:= 1; CLEAR BUSBUSY;
12 01543 4605431640230011           ZAR   OR,WRK3,LAD           CONT WRT   ; LAD:= WRK3, WRT;
13 01544 0655541740000100           XA    AND,CIOST,WRK5        CONT 0 1   ; IF IOSTAT(21:23) <> 0 THEN
14 01545 4631000000421556           JMP   T,NZ,ICER1                       ;   GOTO ICER1;
15 01546 2005240000000024           ZQR   AND,CAMBC             RTN        ; CAMBC:= 0, RETURN;
16                          LOKM2:                                         ;
17 01547 4611730000003724           CZR   OR,.1,CAMBC                      ; CAMBC:= 1;
18 01550 4605240000210011           ZQR   AND,LAD               CONT READ  ; LAD:= 0, READ;
19                          ; START READ IN ORDER TO UNLOCK THE BUS
20 01551 0641622000000700           SQ    SUB0,CUR                         ; IF Q <> CUR THEN
21 01552 4631000000421560           JMP   1,NZ,ICWT                        ;   GOTO ICWT;
22 01553 4613730032002000           CZB   OR,.1817,WRK0                    ; SET NOT-UNLOCK CONDITION;
23 01554 0655730000004100           XZ    OR,IOSTA              CONT 0 1   ; WAIT IO;
24 01555 2005240000000024           ZQR   AND,CAMBC             RTN        ; CAMBC:= 0, RETURN;
25
26                          ICER1:                                         ;
27 01556 4605240000000024           ZQR   AND,CAMBC                        ; CAMBC:= 0;
28 01557 4631000000000034           JMP   F,F,SYFL1                        ; GOTO SYSTEMFAULT1
```

```
10048 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ICWT:                                                ;ICWT:
03 01560 0624730002000200           IZQ   OR,8.,8.                              ; ILIM:= 8;
04 01561 0605230000C00021           ZQR   OR,ILIM                               ; DELAY 900 NSEC BEFORE INTERRUPT TEST
05 01562 0655730000004100           XZ    OR,IOSTA          CONT 0 1            ; SYNC IO
06 01563 4605240000C00024           ZQR   AND,CAMBC                           ; CAMBC:= 0;
07 01564 0607151431000006           ABBR  CAND,.2,CNTR,CNTRO                  ; CNTR(22),CNTRO(22):= 0; SET WAIT LAMP
08 01565 0745230000C00000           ZQ    OR                CONT 0 3            ;
09 01566 0745230000C00000           ZQ    OR                CONT 0 3            ;   DELAY
10                          ICLOP:                                               ; REPEAT
11 01567 0631000002611576           CALL  T,EXINT,ICEXT                       ;   IF EXTINT THEN ICEXT
12 01570 0651513340213011           CAR   SUN0,.6,INF,LAD   CONT READ         ;   LAD:= INF-6, READ;
13 01571 0631000002601614           CALL  T,CPUIN,ICINT                       ;   IF CPUINTERRUPT THEN CALL ICINT;
14                                                                            ; C. ALWAYS ENABLE CPU_INTERRUPT;
15 01572 4655730000004000           XZ    OR,DATI           CONT 0 1            ;   SYNC IO;
16 01573 0631000000421567           JMP   T,NZ,ICLOP                          ; UNTIL DAT1=0;
17 01574 0607131431000006           ABBR  OR,.2,CNTR,CNTRO                    ; CNTR,CNTRO(22):= 1; CLEAR WAITLAMP
18 01575 0631000000001534           JMP   F,F,LOKM1                           ; GOTO LOCKMONITOR
19
20                          ICEXT:                                               ; TEST EXTERNAL INTERRUPT 8
21 01576 0605230000C00022           ZQR   OR,ILEVR                            ; ILEVR:= Q; REQUEST ILEV
22 01577 4705230000C00000           ZQ    OR                CONT 0 2            ; DELAY;
23 01600 4705230000C00000           ZQ    OR                CONT 0 2            ;
24 01601 4724730000007760           IZQ   OR,0,377          CONT 2              ;
25 01602 4614640000C00320           XQQR  AND,ILEV,INTRG                      ; INTRG,Q:= ILEV AND 8'377
26 01603 0651622000C03600           CQ    SUB0,.8                             ; IF Q<>8 THEN
27 01604 6031000000021605           JMP   F,NZ,.+3          RTN               ; RETURN
28 01605 0605230C00C00017           ZQR   OR,RTCR                             ; RTCR:= Q; REQUEST RTC
29 01606 4705230000C00000           ZQ    OR                CONT 0 2            ; DELAY
30 01607 0615730000C00212           XZR   OR,RTC,DATO                         ; DATO:= RTC;
31 01610 0600730000001100           SZQ   OR,PUTAB                            ;
32 01611 4640600000001000           SQQ   ADD,PUIX                            ;
33 01612 0651600000231311           CQR   ADD,.10,LAD       CONT WRT          ; LAD:= PUTAB+PUIX+10, WRT;
34 01613 2055730000000100           XZ    OR,CIOST          RTN 0 1           ; WAIT IO
35
36                          ICINT:                                               ; ONLY AUTOLOAD INTERRUPTS ARE ALLOWED
37 01614 4631000000254002           JMP   F,NOCPA,OCPAL                       ; IF OCPAL THEN GOTO OCPAUTOLOAD;
38 01615 6031000000264003           JMP   F,NREMA,REMAL     RTN               ; IF REMAUTO THEN GOTO REMOTEAUTOLOAD
39                                                                            ; ELSE RETURN
```

```
10049 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          GRTB:                                                   ;GRTB:
03 01616 6007431072000000           ZAB   OR,W1,WRK0              RTN             ; 1: WRK0:= W1;
04 01617 6007431132000000           ZAB   OR,W2,WRK0              RTN             ; 2: WRK0:= W2;
05 01620 2007431172000000           ZAB   OR,W3,WRK0              RTN             ; 3: WRK0:= W3;
06 01621 6007431232000000           ZAB   OR,STAT,WRK0            RTN             ; 4: WRK0:= STAT;
07 01622 2007730032001000           RZB   OR,ICS,WRK0             RTN             ; 5: WRK0:= IC;
08 01623 2007431272000000           ZAB   OR,CAUSE,WRK0           RTN             ; 6: WRK0:= CAUSE;
09 01624 2007431332000000           ZAB   OR,SB,WRK0              RTN             ; 7: WRK0:= SB;
10 01625 0655541740000100           XA    AND,CIOST,WRK5          CONT 0 1        ; 10: IF CIOS1(21:23)<> 0 THEN
11 01626 4631000000421524           JMP   T,NZ,DMPFL                              ;       GOTO DUMPFAULT;
12 01627 0601730000001012           SZR   OR,PUIX,DATO                            ;     DATO:= PUIX;
13 01630 0653501634231111           CABR  ADD,.12,WRK2,WRK2,LAD   CONT WRT        ;     WRK2,LAD:= WRK2+12, WRT;
14 01631 4631000000001500           JMP   F,F,ICAL4                               ;     <* DUMP PU INDEX *>
15                                                                                ;     GOTO ICAL3;
```

```
1005C CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;PROCEDURE ESCAPE
03                          ;*****************
04
05                          ;ESCAP:                                                  ;ESCAPE:   C. Q==CAUSE
06                          .SCAP:                                                   ;
07 01632 0607230025000000           ZQB   OR,CAUSE                                   ; CAUSE:= Q
08 01633 4611541200003100           CA    AND,.1B3,STAT                              ; IF AFTER ESCAPE THEN
09 01634 4631000000021637           JMP   F,NZ,ESC1                                  ;   STAT:=CPUST:= STAT CAND 1B3
10 01635 4613551224003105           CABR  CAND,.1B3,STAT,STAT,CPUST                  ;   C. CLEAR AFTER ESCAPE
11 01636 6005330026000000           ZB    OR,SB                        RTN           ;   SET SB CONDITIONS
12                          ESC1:                                                    ;   RETURN
13 01637 0662731234000600           HZB   OR,LSWAP,STAT,WRK2                       ; WRK2:= 12 EXT 0 CON STAT(0:11)
14 01640 0607141274000000           ABB   AND,CAUSE,WRK2                           ; WRK2:= WRK2 AND CAUSE
15 01641 0624730000001760           IZQ   OR,0,77                                  ;
16 01642 4607041634000000           AQB   AND,WRK2,WRK2                            ; WRK2:= WRK2 AND 8'77
17 01643 0631000000421645           JMP   T,NZ,ESC2                                ; IF WRK2=0 THEN
18 01644 6005330026000000           ZB    OR,SB                        RTN           ;   SET SB CONDITIONS, RETURN
19 01645 0655730000004100 ESC2:     XZ    OR,IOSTA                     CONT 0 1      ; WAIT FOR OPERAND FETCH
20 01646 0611541200002100           CA    AND,.1B16,STAT                             ; IF STAT(16)<>0 THEN
21 01647 0631000000021653           JMP   F,NZ,ESC21                               ; <* USER MODE *>
22 01650 0640501400001200           SAQ   ADD,EXOFS,.2                             ;
23 01651 0641600000210711           SQR   ADD,CUR,LAD                CONT READ     ;  LAD:= CUR+EXOFS+2, READ;
24 01652 0631000000001654           JMP   F,F,ESC22                                ; ELSE
25                          ESC21:                                                   ;
26 01653 0651501340213011           CAR   ADD,.6,INF,LAD             CONT READ     ;  LAD:= INF+6  , READ
27                          ESC22:                                                   ;
28 01654 4613531224003100           CAB   OR,.1B3,STAT,STAT                        ; STAT:= STAT OR 1B3  C. AFTER ESC
29 01655 4624731000177760           IZQ   OR,4000,7777                             ;
30 01656 4607230036000000           ZQB   OR,WRK4                                  ; WRK4:= 8'40007777  C. MASK
31 01657 4645523400001003           RAR   SUB0,ICS,.2,ICD                          ; ICD:= ICS -2
32 01660 0631000000001703           JMP   F,F,EXC3                                 ; GOTO EXC3
```

```
10051 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          INT00:                                          ;INT00:  C. ENTRY FROM TEST
03 01661 0607340025C00000           ZBB   AND,CAUSE                         ; CAUSE:= 0
04 01662 4611541200003200           CA    AND,.1B2,STAT                     ; IF AFTER AM=1 THEN
05 01663 0631000000021672           JMP   F,NZ,EXCEP                        ;   IC:=IC+2
06 01664 4645501400001003           RAR   ADD,ICS,.2,ICD                    ; C. EXCEPTION IN INDIRECT ADDRESS CALC.
07                          .NTO:                                           ;INTO:
08 01665 0607340025000000           ZBB   AND,CAUSE                         ; CAUSE:= 0
09 01666 4631000000001672           JMP   F,F,EXCEP                         ; GOTO EXCEPTION
10
11                          .NTF:                                           ;
12                          ;INTF:                                          ;FLOATING POINT EXCEPTION:
13 01667 4613730025001500           CZB   OR,.4,CAUSE                       ; CAUSE:= .4;
14 01670 4631000000001672           JMP   F,F,EXCEP                         ;
15
16                          INT1:                                           ;INTEGER EXCEPTION:
17 01671 0607431425C00000           ZAB   OR,.2,CAUSE                       ; CAUSE:= 2
18
19                          EXCEP:                                          ;EXCEPTION:
20 01672 0611541200002100           CA    AND,.1B16,STAT                    ; IF STAT(16)<>0 THEN
21 01673 0631000000021677           JMP   F,NZ,EXC1                         ; <* USER MODE *>
22 01674 0600730000001200           SZQ   OR,EXOFS                          ;
23 01675 0641600000210711           SQR   ADD,CUR,LAD          CONT READ    ;  LAD:= CUR+EXOFS, READ;
24 01676 0631000000001700           JMP   F,F,EXC2                          ; ELSE
25                          EXC1:                                           ;
26 01677 4651501340211511           CAR   ADD,.4,INF,LAD       CONT READ    ;  LAD:= INF+4  , READ
27                          EXC2:                                           ;
28 01700 4613551224003100           CAB   CAND,.1B3,STAT,STAT               ; CLEAR AFTER ESCAPE
29 01701 0624731477777760           IZQ   OR,6377,7777                      ; WRK4:= 8'63777777  C. MASK
30 01702 4607230036000000           ZQB   OR,WRK4                           ;
31                          EXC3:                                           ;COMMON EXC AND ESCAPE
32 01703 4613730037003400           CZB   OR,.7,WRK5                        ; WRK5:= 7
33 01704 4655541740004100           XA    AND,IOSTA,WRK5       CONT 0 1     ; IF IOSTATUS(21:23)<>0 THEN
34 01705 0631000000421421           JMP   T,NZ,INTB3                        ;   GOTO INTB3
35 01706 0617730034004000           XZB   OR,DATI,WRK2                      ; WRK2:= DATAIN
36 01707 0631000000421714           JMP   T,NZ,EXC4                         ; IF DUMPADDRESS=0 THEN
```

```
10052 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                                                                          ; BEGIN
03 01710 4651513240003600          CA    SUNO,.8,CAUSE                      ;   IF ESCAPE THEN
04 01711 4631000000411431          JMP   T,NEG,TESTI                        ;     CAUSE:= C  C. CAUSE > 8
05 01712 0607340025000000          ZBB   AND,CAUSE                          ;   GOTO TEST-INT
06 01713 4631000000001431          JMP   F,F,TESTI                          ; END
07                           EXC4:                                          ;
08 01714 4613730037000700          CZB   OR,.15,WRK5                        ; WRK5:= 15
09 01715 0624730000037340          IZQ   OR,0,GRTB1/3-1                     ; Q:= ADDR(GRTB-1)
10 01716 4607431032000000          ZAB   OR,W0,WRK0                         ; WRK0:= W0;
11 01717 4207113434000000          ABB   SUNO,.2,WRK2          PUSH         ; WRK2:= WRK2-2;
12                                                                          ; FOR I:= 0 TO 7 DO
13                                                                          ; BEGIN
14 01720 4605431500000012          ZAR   OR,WRK0,DATO                       ;   DATAOUT:= WRK0
15 01721 4647101434220011          ABBR  ADD,.2,WRK2,LAD       CONT WRTP    ;   WRK2,LAD:= WRK2+2, WRTP;
16 01722 4630202003000000          CXZQQ ADDO,F,F                           ;   WRK0:= GETREG(Q+1), Q:= Q+1
17 01723 4657541775000100          XAB   AND,CIOST,WRK5,WRK3   CONT 0 1     ;   WRK3:= CPUIOSTATUS(20:23)
18                                                                          ;   IF IOSTATUS(20:23)<>0 THEN
19 01724 6631000000421753          JMP   T,NZ,TSTEX            LRTN         ;     GOTO TEST-EX;
20                                                                          ; END
21                           EXC5:                                          ;
22 01725 4657541775000100          XAB   AND,CIOST,WRK5,WRK3   CONT 0 1     ; IF CIOST(21:23)<>0 THEN
23 01726 0631000000421753          JMP   T,NZ,TSTEX                         ;   GOTO TEST-EX;
24 01727 0607340020000000          ZBB   AND,W0                             ; W0:= 0
25 01730 4607431421000000          ZAB   OR,.2,W1                           ; W1:= 2
26 01731 0613730022001500          CZB   OR,.4,W2                           ; W2:= 4
27 01732 0613730023003000          CZB   OR,.6,W3                           ; W3:= 6
28 01733 4607430035000000          ZAB   OR,W,WRK3                          ; WRK3:= W  C. W-FIELD < 1
29 01734 0624730000001760          IZQ   OR,0,77                            ;
30 01735 4607041263000000          AQB   AND,CAUSE,W3                       ; W3:= CAUSE AND 8'77
```

```
10053 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                                                                          ; CAUSE:= CAUSE(0:17) CON 6 EXT 0
03 01736 0635330025000060       SRZBB OR,CAUSE,Z,NL,F,F                     ;
04 01737 0635330025000060       SRZBB OR,CAUSE,Z,NL,F,F                     ;
05 01740 0635330025000060       SRZBB OR,CAUSE,Z,NL,F,F                     ;
06 01741 4613551265003400       CAB   CAND,.7,CAUSE,CAUSE                   ;
07 01742 4635131665000060       SRABB OR,WRK3,CAUSE,Z,NL,F,F                ; CAUSE:= CAUSE SHIFT (-4) + W-FIELD
08 01743 4607730020001000       RZB   OR,ICS,W0                             ; W0:= ICS
09 01744 0607431261000000       ZAB   OR,CAUSE,W1                           ; W1:= CAUSE
10 01745 0607431322000000       ZAB   OR,SB,W2                              ; W2:= SB
11 01746 4647101434240011       ABBR  ADD,.2,WRK2,LAD       CONT READJ      ; LAD,WRK2:= WRK2+2, READ INSTRUCTION;
12 01747 0607141724000005       ABBR  AND,WRK4,STAT,CPUST                   ; STAT:=CPUST:= STAT AND MASK
13 01750 0631000000130032       JMP   F,NWADR,INT0                          ; IF W-ADDR THEN GOTO INT0
14 01751 4605431600000003       ZAR   OR,WRK2,ICD                           ; ICD:= WRK2
15 01752 7455730000610100       XZ    OR,CIOST              NEXT TLIIN 1  ; TEST LIMIT, TEST INT
16                                                                          ; GOTO NEXT INSTR
17                          TSTEX:                                          ;TEST-EX:
18 01753 0611541640003600       CA    AND,.1B20,WRK3                        ; IF IOSTATUS(20)=0 THEN
19 01754 0631000000021413       JMP   F,NZ,INTB2                            ;   GOTO INTB2
20                                                                          ; ELSE
21 01755 0613730025003000       CZB   OR,.6,CAUSE                           ;   CAUSE:= 6
22 01756 4631000000001431       JMP   F,F,TEST1                             ;   GOTO TEST-INT
```

```
10C54 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                              GRTB1:                                                          ;GRTB1:
03 01757 6007431072000000               ZAB   OR,W1,WRK0          RTN                       ; 1: WRK0:= W1;
04 01760 6007431132000000               ZAB   OR,W2,WRK0          RTN                       ; 2: WRK0:= W2;
05 01761 2007431172000000               ZAB   OR,W3,WRK0          RTN                       ; 3: WRK0:= W3;
06 01762 6007431232000000               ZAB   OR,STAT,WRK0        RTN                       ; 4: WRK0:= STAT;
07 01763 2007730032001000               RZB   OR,ICS,WRK0         RTN                       ; 5: WRK0:= IC;
08 01764 2007431272000000               ZAB   OR,CAUSE,WRK0       RTN                       ; 6: WRK0:= CAUSE;
09 01765 2007431332000000               ZAB   OR,SB,WRK0          RTN                       ; 7: WRK0:= SB;
10 01766 4631000000001725               JMP   F,F,EXC5                                      ; GOTO EXC5;
```

```
10055 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;SYSTEMFAULT 0-2
03                          ;****************
04
05                          SYFL0:                                  ; SYSTEMFAULT CAUSE 0
06 01767 0624730000000000           IZQ   OR,0,0                    ; TEST INTERRUPT, ILIM=0
07 01770 4631000000004021           JMP   F,F,SYFLT                 ;
08
09                          ;SYFL1:                                 ; SYSTEMFAULT CAUSE 1
10                          .YFL1:                                  ;
11 01771 4624730000000020           IZQ   OR,0,1                    ; BUSERROR IN FETCH OF NEW STATUS
12 01772 4631000000004021           JMP   F,F,SYFLT                 ; AT INTERRUPT
13
14                          SYFL2:                                  ; SYSTEMFAULT CAUSE 2
15 01773 4624730000000040           IZQ   OR,0,2                    ; DUMPERROR OVERRUN
16 01774 4631000000004021           JMP   F,F,SYFLT                 ;
```

```
10056 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          FILL  2CCO
03                          .EOT
04
05                          ;FILE: T2, EDITED: 850814/LBJ
06
07                    CC0001 .NOMA 1
08                    CC6000 .LOC 20CC*3
09
10                          ;STEPPING STONES
11 C2000 4631000000CC2614          JMP   F,F,.HCLK                    ; GOTO CHECK CLOCK;
12 C2001 0631000000CC2736          JMP   F,F,.ISPA                    ; GOTO DISPATCHING POINT
13
14                          ;ADDRESS CALCULATION ROUTINES
15                          ;*****************************
16
17                          ;THE ADDRESS CALCULATIONS ARE CALLED VIA A SEPARATE TABLE
18                          ;INPUT TO THE TABLE ARE:
19
20                          ;AFTER ESCAPE = 1, ESCAPE = 1
21                          ;*****************************
22
23                          AFTES:                                         ;
24 C2002 3605431300100011          ZAR   OR,SB,LAD             EXEC READC ; LAD:= SB, COND READP
25                                                                         ; GOTO INSTR EXECUTION
26
27                          ;AFTER ESCAPE = 1, ESCAPE = 0
28                          ;*****************************
29
30                          AENES:                                         ;
31 C2003 4613551224003105          CABR  CAND,.1B3,STAT,STAT,CPUST        ; AFTER ESCAPE:= 0
32 C2004 3605431300100011          ZAR   OR,SB,LAD             EXEC READC ; LAD:= SB, COND READP
33                                                                         ; GOTO INSTR EXECUTION
```

```
10057 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;AFTER ESCAPE = 1, AFTER AM = 1
03                          ;*******************************
04
05                          INESC:                                                  ;
06 C2005 0605431300200011           ZAR   OR,SB,LAD              CONT READP     ; LAD:= SB    , READP
07 C2006 4613551224003105           CABR  CAND,.1B3,STAT,STAT,CPUST             ; AFTER ESCAPE:= 0
08 C2007 0631000000132027           JMP   F,NWADR,REGAD                         ; IF W-ADDR THEN GOTO REG. ADDR
09 C2010 4655730000514100           XZ    OR,IOSTA               CONT TSTIO 1 ; TEST IO
10 C2011 4613551224003205           CABR  CAND,.1B2,STAT,STAT,CPUST             ; AFTER AM:= 0
11 C2012 7617730026104011           XZBR  OR,DATI,SB,LAD         EXEC READC     ; SB:=LAD:=DATAIN,  COND READP
12                                                                              ; GOTO INSTR EXEC
13
14                          ;AFTER AM = 0, INDIRECT = 0, EXCAPE = 0
15                          ;**************************************
16
17                          DIR:                                                    ;DIRECT:
18 C2013 7607730026100011           RZBR  OR,DISP,SB,LAD         EXEC READC     ; SB:=LAD:=D,  COND READP
19                                                                              ; GOTO INSTR EXEC
20
21                          REL:                                                    ;RELATIVE:
22 C2014 3607730026102011           RZBR  OR,DIC,SB,LAD          EXEC READC     ; SB:=LAD:= D+IC, COND READP
23                                                                              ; GOTO INSTR EXEC
24
25                          INDEX:                                                  ;INDEX:
26 C2015 3647500126100011           RABR  ADD,DISP,X,SB,LAD      EXEC READC     ; SB:=LAD:= D+X ,  COND READP
27                                                                              ; GOTO INSTR EXEC
28
29                          RELX:                                                   ;RELATIVEINDEX:
30 C2016 7647500126102011           RABR  ADD,DIC,X,SB,LAD       EXEC READC     ; SB:=LAD:= D+IC+X , COND READP
31                                                                              ; GOTO INSTR EXEC
```

```
10058 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;AFTER AM = 0, INDIRECT = 1, ESCAPE = 0
03                          ;**************************************
04
05                          DIRI:                                                   ;
06 C2017 4607730026200011           RZBR   OR,DISP,SB,LAD          CONT READP    ; SB:=LAD:= D  , READP
07 C2020 4631000000132025           JMP    F,NWADR,REGA1                          ; IF W=ADDR THEN GOTO REG ADDR1
08 C2021 4613531224003205           CABR   OR,.1B2,STAT,STAT,CPUST                ; AFTER AM:= 1
09 C2022 4655730000514100           XZ     OR,IOSTA                CONT TSTIO 1  ; TEST IO
10 C2023 4613551224003205           CABR   CAND,.1B2,STAT,STAT,CPUST              ; AFTER AM:= 0
11 C2024 7617730026104011           XZBR   OR,DATI,SB,LAD          EXEC READC    ; SB:=LAD:= DATAIN ,  COND READP
12                                                                                ; GOTO INSTR EXEC
13
14                          REGA1:                                                  ; REG ADR1:
15 C2025 4655730000000100           XZ     OR,CIOST                CONT U 1      ; WAIT IO
16 C2026 7607430166100011           ZABR   OR,LAX,SB,LAD           EXEC READC    ; SB:=LAD:= LAX  ,  COND READP
17                                                                                ; GOTO INSTR EXEC
18
19                          REGAD:                                                  ;REG ADDR:
20 C2027 4655730000000100           XZ     OR,CIOST                CONT U 1      ; WAIT IO
21 C2030 4613551224003205           CABR   CAND,.1B2,STAT,STAT,CPUST              ; AFTER AM:= 0
22 C2031 7607430166100011           ZABR   OR,LAX,SB,LAD           EXEC READC    ; SB=LAD:= LAX  ,  COND READP
23                                                                                ; GOTO INSTR EXEC
24
25                          RELI:                                                   ;
26 C2032 0607730026202011           RZBR   OR,DIC,SB,LAD           CONT READP    ; SB:=LAD:= D+IC  , READP
27 C2033 4631000000132025           JMP    F,NWADR,REGA1                          ; IF W=ADDR THEN GOTO REG ADDR1
28 C2034 4613531224003205           CABR   OR,.1B2,STAT,STAT,CPUST                ; AFTER AM:= 1
29 C2035 4655730000514100           XZ     OR,IOSTA                CONT TSTIO 1  ; TEST IO
30 C2036 4613551224003205           CABR   CAND,.1B2,STAT,STAT,CPUST              ; AFTER AM:= 0
31 C2037 7617730026104011           XZBR   OR,DATI,SB,LAD          EXEC READC    ; SB:=LAD:= DATAIN  , COND READP
32                                                                                ; GOTO INSTR EXEC
```

```
10059 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          INDXI:                                          ;
03 C2040 0647500126200011           RABR  ADD,DISP,X,SB,LAD      CONT READP   ; SB:=LAD:= D+X  ,  READP
04 C2041 4631000000132025           JMP   F,NWADR,REGA1                       ; IF W-ADDR THEN GOTO REG ADDR1
05 C2042 4613531224003205           CABR  OR,.1B2,STAT,STAT,CPUST             ; AFTER AM:= 1
06 C2043 4655730000514100           XZ    OR,IOSTA               CONT TSTIO 1 ; TEST IO
07 C2044 4613551224003205           CABR  CAND,.1B2,STAT,STAT,CPUST           ; AFTER AM:= 0
08 C2045 7617730026104011           XZBR  OR,DATI,SB,LAD         EXEC READC   ; SB:=LAD:= DATAIN  , COND READP
09                                                                            ; GOTO INSTR EXEC
10
11                          RELXI:                                          ;
12 C2046 4647500126202011           RABR  ADD,DIC,X,SB,LAD      CONT READP    ; SB:=LAD:= D+IC+X  , READP
13 C2047 4631000000132025           JMP   F,NWADR,REGA1                       ; IF W-ADDR THEN GOTO REG ADDR1
14 C2050 4613531224003205           CABR  OR,.1B2,STAT,STAT,CPUST             ; AFTER AM:= 1
15 C2051 4655730000514100           XZ    OR,IOSTA               CONT TSTIO 1 ; TESTIO
16 C2052 4613551224003205           CABR  CAND,.1B2,STAT,STAT,CPUST           ; AFTER AM:= 0
17 C2053 7617730026104011           XZBR  OR,DATI,SB,LAD         EXEC READC   ; SB:=LAD:= DATAIN  , COND READP
18                                                                            ; GOTO INSTR EXEC
19
20                          ;AFTER AM = 0, INDIRECT = 1, ESCAPE = 1
21                          ;**************************************
22
23                          DIRIE:                                          ;
24 C2054 4607730026200011           RZBR  OR,DISP,SB,LAD         CONT READP   ; SB:=LAD:= D  ,  READP
25 C2055 4613531224003205           CABR  OR,.1B2,STAT,STAT,CPUST             ; AFTER AM:= 1
26 C2056 0631000002002167           CALL  F,F,ESCA1                           ; CALL ADDR ESCAPE(10002)
27 C2057 0631000000132027           JMP   F,NWADR,REGAD                       ; IF W-ADDR THEN GOTO REG ADDR
28 C2060 4655730000514100           XZ    OR,IOSTA               CONT TSTIO 1 ; TEST IO
29 C2061 4613551224003205           CABR  CAND,.1B2,STAT,STAT,CPUST           ; AFTER AM:= 0
30 C2062 7617730026104011           XZBR  OR,DATI,SB,LAD         EXEC READC   ; SB:=LAD:= DATAIN  , COND READP
31                                                                            ; GOTO INSTR EXEC
```

```
1006C CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          RELIE:                                          ;
03 C2063 0607730026202011          RZBR  OR,DIC,SB,LAD          CONT READP  ; SB:=LAD:= D+IC  , READP
04 C2064 4613531224003205          CABR  OR,.1B2,STAT,STAT,CPUST            ; AFTER AM:= 1
05 C2065 0631000002002167          CALL  F,F,ESCA1                          ; CALL ADDR ESCAPE(10002)
06 C2066 0631000000132027          JMP   F,NWADR,REGAD                      ; IF W-ADDR THEN GOTO REG ADDR
07 C2067 4655730000514100          XZ    OR,IOSTA               CONT TSTIO 1; TEST IO
08 C2070 4613551224003205          CABR  CAND,.1B2,STAT,STAT,CPUST          ; AFTER AM:= 0
09 C2071 7617730026104011          XZBR  OR,DATI,SB,LAD         EXEC READC  ; SB:=LAD:= DATAIN  , COND READP
10                                                                          ; GOTO INSTR EXEC
11
12                          IDXIE:                                          ;
13 C2072 0647500126200011          RABR  ADD,DISP,X,SB,LAD      CONT READP  ; SB:=LAD:= D+X  , READP
14 C2073 4613531224003205          CABR  OR,.1B2,STAT,STAT,CPUST            ; AFTER AM:= 1
15 C2074 0631000002002167          CALL  F,F,ESCA1                          ; CALL ADDR ESCAPE(10002)
16 C2075 0631000000132027          JMP   F,NWADR,REGAD                      ; IF W-ADDR THEN GOTO REG ADDR
17 C2076 4655730000514100          XZ    OR,IOSTA               CONT TSTIO 1; TEST IO
18 C2077 4613551224003205          CABR  CAND,.1B2,STAT,STAT,CPUST          ; AFTER AM:= 0
19 C2100 7617730026104011          XZBR  OR,DATI,SB,LAD         EXEC READC  ; SB:=LAD:= DATAIN  , COND READP
20                                                                          ; GOTO INSTR EXEC
21
22                          RLXIE:                                          ;
23 C2101 4647500126202011          RABR  ADD,DIC,X,SB,LAD       CONT READP  ; SB:=LAD:= D+IC+X  , READP
24 C2102 4613531224003205          CABR  OR,.1B2,STAT,STAT,CPUST            ; AFTER AM:= 1
25 C2103 0631000002002167          CALL  F,F,ESCA1                          ; CALL ADDR ESCAPE(10002)
26 C2104 0631000000132027          JMP   F,NWADR,REGAD                      ; IF W-ADDR THEN GOTO REG ADDR
27 C2105 4655730000514100          XZ    OR,IOSTA               CONT TSTIO 1; TEST IO
28 C2106 4613551224003205          CABR  CAND,.1B2,STAT,STAT,CPUST          ; AFTER AM:= 0
29 C2107 7617730026104011          XZBR  OR,DATI,SB,LAD         EXEC READC  ; SB:=LAD:= DATAIN  , COND READP
30                                                                          ; GOTO INSTR EXEC
```

```
10061 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;AFTER AM = 1, INDIRECT = 0, ESCAPE = 0
03                          ;**************************************
04
05                          AMR:                                                    ;
06 C2110 4613551224003205           CABR  CAND,,1B2,STAT,STAT,CPUST                 ; AFTER AM:= 0
07 C2111 7647501326102011           RABR  ADD,DIC,SB,SB,LAD      EXEC READC         ; SB:= SB+IC+D, COND READP
08                                                                                  ; GOTO INSTR EXEC
09 C2112 4613551224003205 AMD:      CABR  CAND,,1B2,STAT,STAT,CPUST                 ; AFTER AM:= 0
10 C2113 3647501326100011           RABR  ADD,DISP,SB,SB,LAD     EXEC READC         ; SB:= LAD:= D+ SB , COND READP
11                                                                                  ; GOTO INSTR EXEC
12
13                          AMRX:                                                   ;
14 C2114 4613551224003205           CABR  CAND,,1B2,STAT,STAT,CPUST                 ; AFTER AM:= 0
15 C2115 0647501326002000           RAB   ADD,DIC,SB,SB                             ; SB:= SB+IC+D
16 C2116 7647100126100011           ABBR  ADD,X,SB,LAD           EXEC READC         ; SB:= SB+X , COND READP
17                                                                                  ; GOTO INSTR EXEC
18 C2117 4613551224003205 AMX:      CABR  CAND,,1B2,STAT,STAT,CPUST                 ; AFTER AM:= 0
19 C2120 4647501326000000           RAB   ADD,DISP,SB,SB                            ; SB:= D+SB
20 C2121 7647100126100011           ABBR  ADD,X,SB,LAD           EXEC READC         ; SB:=LAD:= SB+X ,  COND READP
21                                                                                  ; GOTO INSTR EXEC
```

```
10062 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;AFTER AM = 1, INDIRECT = 1, ESCAPE = 0
03                          ;**************************************
04
05                          AMRI:                                                ;
06 C2122 4647501326202011          RABR   ADD,DIC,SB,SB,LAD      CONT READP    ; SB,LAD:= SB+IC+C, READP;
07 C2123 0631000000132027          JMP    F,NWADR,REGAD                        ; IF W-ADDR THEN GOTO REG ADDR;
08 C2124 4655730000514100          XZ     OR,IOSTA               CONT TSTIO 1  ; TEST IO
09 C2125 4613551224003205          CABR   CAND,.1B2,STAT,STAT,CPUST            ; AFTER AM:= 0;
10 C2126 7617730026104011          XZBR   OR,DATI,SB,LAD         EXEC READC    ; SB,LAD:= DATI, COND READP;
11                                                                              ; GOTO INSTR EXEC
12
13                          AMDI:                                                ;
14 C2127 0647501326200011          RABR   ADD,DISP,SB,SB,LAD     CONT READP    ; SB:=LAD:= D+SB  , READP
15 C2130 0631000000132027          JMP    F,NWADR,REGAD                        ; IF W-ADDR THEN GOTO REG ADDR
16 C2131 4655730000514100          XZ     OR,IOSTA               CONT TSTIO 1  ; TEST IO
17 C2132 4613551224003205          CABR   CAND,.1B2,STAT,STAT,CPUST            ; AFTER AM:= 0
18 C2133 7617730026104011          XZBR   OR,DATI,SB,LAD         EXEC READC    ; SB:=LAD:= DATAIN  ,  COND READP
19                                                                              ; GOTO INSTR EXEC
20
21                          AMRXI:                                               ;
22 C2134 0647501326002000          RAB    ADD,DIC,SB,SB                        ; SB:= SB+IC+D;
23 C2135 4647100126200011          AEBR   ADD,X,SB,LAD           CONT READP    ; SB,LAD:= SB+X, READP;
24 C2136 0631000000132027          JMP    F,NWADR,REGAD                        ; IF W-ADDR THEN GOTO REG ADDR;
25 C2137 4655730000514100          XZ     OR,IOSTA               CONT TSTIO 1  ; TEST IO;
26 C2140 4613551224003205          CABR   CAND,.1B2,STAT,STAT,CPUST            ; AFTER AM:= 0;
27 C2141 7617730026104011          XZBR   OR,DATI,SB,LAD         EXEC READC    ; SB,LAD:= DATI, COND READP
28                                                                              ; GOTO INSTR EXEC
29
30                          AMXI:                                                ;
31 C2142 0644500100000000          RAQ    ADD,DISP,X                           ; Q:= D+X
32 C2143 0647001326200011          AQBR   ADD,SB,SB,LAD          CONT READP    ; SB:=LAD:= Q+SB  ,  READP
33 C2144 0631000000132027          JMP    F,NWADR,REGAD                        ; IF W-ADDR THEN GOTO REGADR
34 C2145 4655730000514100          XZ     OR,IOSTA               CONT TSTIO 1  ; TEST IO
35 C2146 4613551224003205          CABR   CAND,.1B2,STAT,STAT,CPUST            ; AFTER AM:= 0
36 C2147 7617730026104011          XZBR   OR,DATI,SB,LAD         EXEC READC    ; SB:=LAD:= DATAIN  , COND READP
37                                                                              ; GOTO INSTR EXEC
```

```
10063 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;AFTER AM = 1, INDIRECT = 1, ESCAPE = 1
03                          ;**************************************
04
05                          AMRIE:                                                  ;
06 C2150 0647501326CC1000           RAB    ADD,ICS,SB,SB                         ; SB:= SB+ICS
07 C2151 0647501326200011 AMDIE:    RABR   ADD,DISP,SB,SB,LAD     CONT READP    ; SB:=LAD:= SB+D  , READP
08 C2152 0631000002C02167           CALL   F,F,ESCA1                             ; CALL ADDR ESCAPE(10002)
09 C2153 0631000000132027           JMP    F,NWADR,REGAD                         ; IF W-ADDR THEN GOTO REG ADDR
10 C2154 4655730000514100           XZ     OR,IOSTA               CONT TSTIO 1  ; TEST IO
11 C2155 4613551224003205           CABR   CAND,,1B2,STAT,STAT,CPUST             ; AFTER AM:= C
12 C2156 7617730026104011           XZBR   OR,DATI,SB,LAD         EXEC READC    ; SB:=LAD:= DATAIN  , COND READP
13                                                                               ; GOTO INSTR EXEC
14
15                          ARXIE:                                                  ;
16 C2157 0647501326CC1000           RAB    ADD,ICS,SB,SB                         ; SB:= SB+ICS
17 C2160 0644500100CC0000 AMXIE:    RAQ    ADD,DISP,X                            ; Q:= D+X
18 C2161 0647001326200011           AQBR   ADD,SB,SB,LAD          CONT READP    ; SB:=LAD:= SB+Q
19 C2162 0631000002C02167           CALL   F,F,ESCA1                             ; CALL ADDR ESCAPE(10002)
20 C2163 0631000000132027           JMP    F,NWADR,REGAD                         ; IF W-ADDR THEN GOTO REG ADDR
21 C2164 4655730000514100           XZ     OR,IOSTA               CONT TSTIO 1  ; TEST IO
22 C2165 4613551224003205           CABR   CAND,,1B2,STAT,STAT,CPUST             ; AFTER AM:= C
23 C2166 7617730026104011           XZBR   OR,DATI,SB,LAD         EXEC READC    ; SB:=LAD:= DATAIN  , COND READP
24                                                                               ; GOTO INSTR EXEC
25
26                          ESCA1:                                                  ;ESCAPE IN INDIRECT ADDRESS CALC
27 C2167 4645501400C01003           RAR    ADD,ICS,,2,ICD                        ; IC:= IC+2
28 C2170 0624730000200040           IZQ    OR,1,2                                ;
29 C2171 0631000002C00031           CALL   F,F,ESCAP                             ; ESCAPE(10002)
30 C2172 2045523400CC1003           RAR    SUBO,ICS,,2,ICD       RTN            ; IC:= IC-2
```

```
10064 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;SKIP INSTRUCTIONS
03                          ;******************
04
05 C2173 7431000000002174 SKIP:  JMP   F,F,.+3                NEXT          ; INCREASE PC
06 C2174 4645501400001003         RAR   ADD,ICS,.2,ICD                        ; ICD:= ICS+2
07 C2175 7405230000600000         ZQ    OR                      NEXT TSTIN    ; TEST INT, GOTO NEXT INSTR
08
09                          ;SKIP IF REGISTER HIGH, SH, OPCODE = 50
10
11                          SHE:                                              ;SHE:
12 C2176 0624730000120420         IZQ   OR,0,5021                             ;
13 C2177 0631000002C00031         CALL  F,F,ESCAP                             ; CALL ESCAPE(5021)
14                          SH:                                               ;SH:
15 C2200 4605123300C00000         AB    SUBO,SB,W                           ; IF W > SB THEN
16 C2201 0631000000452173         JMP   T,SIGN,SKIP                         ;   GOTO SKIP
17 C2202 7405230000600000         ZQ    OR                      NEXT TSTIN    ; TEST INT
18                                                                          ; GOTO NEXT INSTR
19
20                          ;SKIP IF REGISTER LOW, SL, OPCODE = 51
21
22                          SLE:                                              ;SLE:
23 C2203 4624730000122420         IZQ   OR,0,5121                             ;
24 C2204 0631000002C00031         CALL  F,F,ESCAP                             ; CALL ESCAPE(5121)
25                          SL:                                               ;SL:
26 C2205 4605113300C00000         AB    SUNO,SB,W                           ; IF W < SB THEN
27 C2206 0631000000452173         JMP   T,SIGN,SKIP                         ;   GOTO SKIP
28 C2207 7405230000600000         ZQ    OR                      NEXT TSTIN    ; TEST INT
29                                                                          ; GOTO NEXT INSTR
30
31                          ;SKIP IF REGISTER EQUAL, SE, OPCODE = 52
32
33                          SEE:                                              ;SEE:
34 C2210 4624730000124420         IZQ   OR,0,5221                             ;
35 C2211 0631000002C00031         CALL  F,F,ESCAP                             ; CALL ESCAPE(5221X)
36                          SE:                                               ;SE:
37 C2212 4605123300C00000         AB    SUBO,SB,W                           ; IF W = SB THEN
38 C2213 0631000000C22173         JMP   F,NZ,SKIP                           ;   GOTO SKIP
39 C2214 7405230000600000         ZQ    OR                      NEXT TSTIN    ; TEST INT
40                                                                          ; GOTO NEXT INSTR
```

```
10065 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;SKIP IF REGISTER NOT EQUAL, SN, OPCODE = 53
03
04                          SNE:                                                ;SNE:
05 C2215 0624730000126420           IZQ   OR,0,5321                           ;
06 C2216 0631000002C00031           CALL  F,F,ESCAP                           ; CALL ESCAPE(5321)
07                          SN:                                                 ;SN:
08 C2217 4605123300CC0000           AB    SUB0,SB,W                           ; IF W <> SB THEN
09 C2220 4631000000422173           JMP   T,NZ,SKIP                           ;   GOTO SKIP
10 C2221 7405230000600000           ZQ    OR                    NEXT TSTIN    ; TEST INT
11                                                                            ; GOTO NEXT INSTR
12
13                          ;SKIP IF REGISTER BITS ONE, SO, OPCODE = 54
14
15                          SOE:                                                ;SOE:
16 C2222 4624730000130420           IZQ   OR,0,5421                           ;
17 C2223 0631000002000031           CALL  F,F,ESCAP                           ; CALL ESCAPE(5421)
18                          SO:                                                 ;SO:
19 C2224 0607130026000000           ABB   OR,W,SB                             ; IF W = W OR SB THEN
20 C2225 4605123300CC0000           AB    SUB0,SB,W                           ;   GOTO SKIP
21 C2226 0631000000022173           JMP   F,NZ,SKIP                           ;
22 C2227 7405230000600000           ZQ    OR                    NEXT TSTIN    ; TEST INT
23                                                                            ; GOTO NEXT INSTR
24
25                          ;SKIP IF REGISTER BITS ZERO, SZ, OPCODE = 55
26
27                          SZE:                                                ;SZE:
28 C2230 0624730000132420           IZQ   OR,0,5521                           ;
29 C2231 0631000002C00031           CALL  F,F,ESCAP                           ; CALL ESCAPE(5521)
30                          SZERO:                                              ;SZ:
31 C2232 0605141300CC0000           AB    AND,SB,W                            ; IF W AND SB = 0 THEN
32 C2233 0631000000022173           JMP   F,NZ,SKIP                           ;   GOTO SKIP
33 C2234 7405230000600000           ZQ    OR                    NEXT TSTIN    ; TEST INT
34                                                                            ; GOTO NEXT INSTR
```

```
10066 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;SKIP IF NO EXCEPTIONS, SX, OPCODE = 56
03
04                          SXE:                                                ;SXE:
05 C2235 0624730000134420           IZQ   OR,0,5621                           ;
06 C2236 0631000002000031           CALL  F,F,ESCAP                           ; CALL ESCAPE(5621)
07                          SX:                                                 ;SX:
08 C2237 0610541200003400           CAQ   AND,.7,STAT                         ; Q:= STAT(21:23)
09 C2240 4605041300000000           AQ    AND,SB                              ; IF SB AND Q = 0 THEN
10 C2241 0631000000022173           JMP   F,NZ,SKIP                           ;   GOTO SKIP
11 C2242 7405230000600000           ZQ    OR                    NEXT TSTIN    ; TEST INT
12                                                                            ; GOTO NEXT INSTR
13
14                          ;SKIP IF NO WRITE PROTECTION, SP, OPCODE = 25
15
16                          SPE:                                                ;SPE:
17 C2243 4624730000052420           IZQ   OR,0,2521                           ;
18 C2244 0631000002000031           CALL  F,F,ESCAP                           ; CALL ESCAPE(2521)
19                          SP:                                                 ;SP:
20 C2245 4631000000412254           JMP   T,NEG,SPEXI                         ; IF SB < 0 THEN
21                                                                            ;   TEST INT, GOTO NEXT INSTR
22 C2246 0643501333000100           SAB   ADD,BASEC,SB,WRK1                   ; WRK1:=SB + BASEC
23 C2247 0641513540000200           SA    SUNO,LLIMC,WRK1                     ; IF SB + BASEC < LOWER LIMIT THEN
24 C2250 0631000000412253           JMP   T,NEG,SP2                           ;   GOTO SP2
25 C2251 4641513540000300           SA    SUNO,ULIMC,WRK1                     ; IF SB + BASEC < UPPER LIMIT THEN
26 C2252 4631000000412173           JMP   T,NEG,SKIP                          ;   GOTO SKIP
27 C2253 0631000000132173 SP2:      JMP   F,NWADR,SKIP                        ; IF SB<8 THEN GOTO SKIP
28 C2254 7405230000600000 SPEXI:    ZQ    OR                    NEXT TSTIN    ; TEST INT, GOTO NEXT INSTR
29
```

```
10067 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;INPUT-OUTPUT INSTRUCTIONS
03                          ;*************************
04
05                          ;DATA IN, DI, OPCODE = 35, PRIVILEGED INSTRUCTION
06
07                          DIE:                                                ;DIE:
08 02255 0624730000073020           IZQ   OR,0,3541                           ;
09 02256 0631000002000031           CALL  F,F,ESCAP                           ; CALL ESCAPE(3541)
10                          DI:                                                 ;DI:
11 02257 0631000000160032           JMP   F,MMODE,INTO                        ; IF -,MONMODE THEN GOTO INTC
12 02260 4605431300210011           ZAR   OR,SB,LAD             CONT READ     ; LAD:=SB,   READ
13 02261 4613730032003400           CZB   OR,.7,WRKO                          ; WRKO:= 7
14 02262 0607151524000000           ABB   CAND,WRKO,STAT                      ; STAT:= STAT(0:20) CON 3 EXT 0
15 02263 0631000000542267           JMP   T,NGADR,DIDEV                       ; IF MEMORY ADDRESS THEN
16                                                                            ; BEGIN
17 02264 4657541532004100           XAB   AND,IOSTA,WRKO,WRKO   CONT 0 1      ;   WRKO:= 21 EXT 0 CON IOSTATUS(21:23)
18 02265 4617730000004000           XZB   OR,DATI,W                           ;   W:= DATAIN
19 02266 7407131524600000           ABB   OR,WRKO,STAT          NEXT TSTIN    ; END ELSE
20                          DIDEV:                                            ; BEGIN
21 02267 0657541532000100           XAB   AND,CIOST,WRKO,WRKO   CONT 0 1      ;   WRKO:=21 EXT 0 CON CPUIOSTATUS(21:23)
22 02270 0617730000000000           XZB   OR,CDATI,W                          ;   W:= CPUDATAIN
23                                                                            ; END
24 02271 7407131524600000           ABB   OR,WRKO,STAT          NEXT TSTIN    ; STAT:= STAT OR WRKO, TEST INT
25                                                                            ; GOTO NEXT INSTR
26
27                          ;DATA OUT, DO, OPCODE = 1, PRIVILEGED INSTRUCTION
28
29                          DOE:                                              ;DOE:
30 02272 4624730000C03020           IZQ   OR,0,0141                           ;
31 02273 0631000002C00031           CALL  F,F,ESCAP                           ; CALL ESCAPE(0141)
32                          DO:                                               ;DO:
33 02274 0631000000160032           JMP   F,MMODE,INTO                        ; IF -,MONMODE THEN GOTO INTC
34 02275 0605430000C00012           ZAR   OR,W,DATO                           ; DATAOUT:= W
35 02276 0605431300230011           ZAR   OR,SB,LAD             CONT WRT      ; LAD:= SB    ,WRT
36 02277 4613730032003400           CZB   OR,.7,WRKO                          ; WRKO:=7
37 02300 0607151524000000           ABB   CAND,WRKO,STAT                      ; STAT:= STAT(0:20) CON 3 EXT 0
38 02301 0657541532000100           XAB   AND,CIOST,WRKO,WRKO   CONT 0 1      ; WRKO:= 21 EXT 0 CON IOSTATUS(21:23)
39 02302 7407131524600000           ABB   OR,WRKO,STAT          NEXT TSTIN    ; STAT:= STAT OR WRKO, TEST INT
40                                                                            ; GOTO NEXT INSTR
```

```
10068 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;ESCAPE ON UNASSIGNED OPERATION CODES
03                          ;************************************
04
05 C2303 4624730000C01000 F00E:  IZQ   OR,0,0040                    ; OPCODE=0
06 C2304 4631000000C02312        JMP   F,F,ESUNA                    ; GOTO ESUNA
07 C2305 4624730000147000 F63E:  IZQ   OR,0,6340                    ; OPCODE=63
08 C2306 4631000000C02312        JMP   F,F,ESUNA                    ; GOTO ESUNA
09 C2307 0624730000175000 F76E:  IZQ   OR,0,7640                    ; OPCODE=76
10 C2310 4631000000C02312        JMP   F,F,ESUNA                    ; GOTO ESUNA
11 C2311 4624730000177000 F77E:  IZQ   OR,0,7740                    ; OPCODE=77
12 C2312 0631000002C00031 ESUNA: CALL  F,F,ESCAP                    ; CALL ESCAPE
13 C2313 4631000000C00032        JMP   F,F,INTU                     ; GOTO INTU
```

```
10069 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;LOCK SEMAPHORE, LK, OPCODE = 74, PRIVILEGED INSTRUCTION
03
04                          LKE:                                                ;LKE:
05 C2314 4624730000171140           IZQ   OR,0,7446                           ;
06 C2315 0631000002000031           CALL  F,F,ESCAP                           ; CALL ESCAPE(7446);
07
08                          LK:                                                 ;LK:
09 C2316 0631000000160032           JMP   F,MMODE,INT0                        ; IF -,MONMODE THEN GOTO INT0
10                                                                            ; LOCKED:= TRUE;
11 C2317 4613730037000700           CZB   OR,.15,WRK5                         ; WHILE LOCKED DO
12 C2320 0611730000001424 WSEM1:    CZR   OR,.3,CAMBC                         ;  SET CAMBYPASS AND BUSBUSY;
13 C2321 0605431300200011           ZAR   OR,SB,LAD             CONT READP    ;  LAD:= SB , READP
14 C2322 4631000000132362           JMP   F,NWADR,LER2                        ;  IF W-ADDRESS THEN GOTO LER2
15 C2323 0601730000000712           SZR   OR,CUR,DATO                         ;  DATO:= CUR;
16 C2324 4657730032504000           XZB   OR,DATI,WRK0          CONT TSTBE 1  ;  WRK0:= DATI, TEST BUSERROR;
17 C2325 4631000000422336           JMP   T,NZ,WSEM3                          ;  IF Q <> 0 THEN GOTO WSEM3;
18 C2326 4611730000003724           CZR   OR,.1,CAMBC                         ;  CAMBC:= 1; LOCKED:= FALSE;
19 C2327 4605431300220011           ZAR   OR,SB,LAD             CONT WRTP     ;  LAD:= SB, WRTP;
20 C2330 4654541740000100 WSEM2:    XAQ   AND,CIOST,WRK5        CONT 0 1      ;  WAIT IO;
21 C2331 4631000000422355           JMP   T,NZ,LER1                           ;  IF IOSTAT(20:23)<>0 THEN
22                                                                            ;   GOTO LER1;
23 C2332 4611531200003605           CAR   OR,.1B20,STAT,CPUST                 ;  CPUST:= STAT OR 1B20;
24                                                                            ;  <* SELECT DISABLE LIMIT *>
25 C2333 4605240000000024           ZQR   AND,CAMBC                           ;  CLEAR BYPASSCAM;
26 C2334 4605730000241011           RZR   OR,ICS,LAD            CONT READJ    ;  LAD:= ICS, START PREFETCH;
27 C2335 7545230000600000           ZQ    OR                    NEXT TSTIN 3  ;  NEXT INST;
```

```
1007C CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          WSEM3:                                                ;WSEM3:
03 C2336 4611730000003724           CZR   OR,.1,CAMBC                           ;  CLEAR BUSBUSY;
04 C2337 4605240000210011           ZQR   AND,LAD                 CONT READ     ;  LAD:= 0, READ;
05 C2340 4641523500CC0700           SA    SUB0,CUR,WRK0                         ;  IF CUR = WRK0 THEN
06                                                                              ;    LOCKED:= FALSE;
07 C2341 0631000000C22330           JMP   F,NZ,WSEM2                            ;    GOTO WSEM2;
08 C2342 0655730000004100           XZ    OR,IOSTA                CONT 0 1      ;  WAIT IO;
09 C2343 4605240000000024           ZQR   AND,CAMBC                             ;  CLEAR CAMBYPASS;
10                          LOP:                                                ;  WHILE WRK0<>0 DO
11 C2344 0605431300200011           ZAR   OR,SB,LAD               CONT READP    ;    LAD:= SB, READP;
12 C2345 4645523400001003           RAR   SUB0,ICS,.2,ICD                       ;    IC:= IC-2;
13 C2346 0607131431600006           ABBR  OR,.2,CNTR,CNTR0        CONT TSTIN    ;    CLEAR WAIT LAMP, TEST INT;
14 C2347 0607151431000006           ABBR  CAND,.2,CNTR,CNTR0                    ;    SET WAIT LAMP;
15                          ; IF AN INTERRUPT IS PENDING THE INSTRUCTION WILL BE EXECUTED
16                          ; AFTER THE INTERRUT IS SERVICED
17 C2350 4645501400001003           RAR   ADD,ICS,.2,ICD                        ;    IC:=IC+2;
18 C2351 0655730000514000           XZ    OR,DATI                 CONT TSTIO 1  ;    TESTIO;
19 C2352 4631000000422344           JMP   T,NZ,LOP                              ;  END WHILE WRK0<>0
20 C2353 0607131431C00006           ABBR  OR,.2,CNTR,CNTR0                      ;  CLEAR WAIT LAMP
21 C2354 0631000000002320           JMP   F,F,WSEM1                             ; END WHILE LOCKED;
22
23 C2355 4605240000000024 LER1:     ZQR   AND,CAMBC                             ; CLEAR BYPASSCAM;
24 C2356 0611640000C03600           CQ    AND,.1820                             ; IF IOSTAT(20)<>0 THEN
25 C2357 4631000000420032           JMP   T,NZ,INT0                             ; GOTO INT0 ;LIMIT VIOLATION;
26 C2360 0607230032C00000           ZQB   OR,WRK0                               ;
27 C2361 0631000000CC0033           JMP   F,F,INTB1                             ; ELSE GOTO INTB1 ;IO ERROR;
28                          LER2:                                               ;
29 C2362 4655730000CC0100           XZ    OR,CIOST                CONT 0 1      ;
30 C2363 4605240000C00024           ZQR   AND,CAMBC                             ;  CAMBC:= 0;
31 C2364 4631000000000032           JMP   F,F,INT0                              ;  GOTO INT0;
```

```
10071 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;UNLOCK SEMAPHORE, UL, OPCODE = 37 , PRIVILEGED INSTRUCTION
03
04                          ULE:                                                    ;ULE:
05 02365 0624730000077140           IZQ    OR,0,3746                            ;
06 02366 0631000002000031           CALL   F,F,ESCAP                            ; CALL ESCAPE(3746);
07
08                          UL:                                                     ;UL:
09 02367 0631000000160032           JMP    F,MMODE,INTO                         ; IF -,MONMODE THEN GOTO INTO;
10 02370 0631000000130032           JMP    F,NWADR,INTO                         ; IF SB < 8 THEN GOTO INTO;
11 02371 4654730000514000           XZQ    OR,DATI                 CONT TSTIO 1 ; Q:= WORD(SB);
12 02372 0641622000000700           SQ     SUBO,CUR                             ; IF Q <> CUR THEN
13 02373 0631000000422400           JMP    T,NZ,ULEX                            ;   GOTO ULEXIT;
14                          ; DO NOT OPEN IF THE SEMAPHORE WAS NOT LOCED BY THE PROCESS ITSELF
15 02374 0605431200000005           ZAR    OR,STAT,CPUST                        ; CPUST:= STAT; RESTORE CPUST
16                          ; DELAY 495 NSEC BEFORE TESTINT
17 02375 4605240000000012           ZQR    AND,DATO                             ; DATO:= 0;
18 02376 4605431300220011           ZAR    OR,SB,LAD               CONT WRTP    ; LAD:= SB, WRTP;
19 02377 3455730000710100           XZ     OR,CIOST                NEXT IIOIN 1 ; TEST IO, TEST INT, NEXT;
20 02400 7405230000600000 ULEX:     ZQ     OR                      NEXT TSTIN   ;
```

```
10072 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;ACTIVATE PROCESS, AP, OPCODE = 36, PRIVILEGED INSTRUCTION
03
04                          APE:                                                    ;APE:
05 C2401 0624730000075640           IZQ    OR,0,3672                               ;
06 C2402 0631000002C00031           CALL   F,F,ESCAP                               ; CALL ESCAPE(3672);
07                          AP:                                                     ;AP:
08 C2403 0631000000160032           JMP    F,MMODE,INTO                            ; IF -,MONMODE THEN GOTO INTO
09 C2404 4603730026211111           SZBR   OR,PUTAB,SB,LAD       CONT READ         ; SB,LAD:= PUTAB, READ
10 C2405 0643501436001100           SAB    ADD,PUTAB,.2,WRK4                       ; PUSTART:= PUTAB+2;
11 C2406 4624730000CC0640           IZQ    OR,0,26.                                ; Q:= DUMP OFFSET + 26;
12 C2407 4640600000001300           SQQ    ADD,DMOFS                               ; <* OFFSET OF PU-INDEX *>
13 C2410 0657730033504000           XZB    OR,DATI,WRK1          CONT TSTBE 1      ; FREE:= PU_TABLE(U), TESTBE;
14 C2411 4631000000022444           JMP    F,NZ,APEX                               ; IF FREE=0 THEN GOTO EXIT;
15 C2412 0647000026210011           AQBR   ADD,W,SB,LAD          CONT READ         ; SB,LAD:= W + Q, READ
16 C2413 0607431432000000           ZAB    OR,.2,WRK0                              ; WRK0:= 2;
17 C2414 4613730037003400           CZB    OR,.7,WRK5                              ; WRK5:= 7;
18 C2415 0654730000504000           XZQ    OR,DATI               CONT TSTBE 1      ; IX:= DATI, TESTBE;
19 C2416 0631000000422421           JMP    T,NZ,AP1                                ; IF IX<>0 OR
20 C2417 4651513540003700           CA     SUNO,.1,WRK1                            ; FREE=1 THEN
21 C2420 0631000000422435           JMP    T,NZ,AP3                                ; BEGIN
22 C2421 0647001726210011 AP1:      AQBR   ADD,WRK4,SB,LAD       CONT READ         ;  LAD,SB:=PUSTART+IX*2, READ;
23 C2422 4655730000504000           XZ     OR,DATI               CONT TSTBE 1      ;  TESTBE;
24 C2423 0631000000422435           JMP    T,NZ,AP3                                ;  IF DATI=0 THEN
25                          AP2:                                                    ;  BEGIN
26 C2424 0605430000000012           ZAR    OR,W,DATO                               ;   DATO:= W;
27 C2425 0605431300230011           ZAR    OR,SB,LAD             CONT WRT          ;   LAD:= SB, WRT;
28 C2426 4655730000500100           XZ     OR,CIOST              CONT TSTBE 1      ;   TESTBE;
```

```
10073 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 C2427 0645411540CC0012        ZAR   SUN,WRK1,DATO                          ;   DATO:= FREE-1;
03 C2430 0603730026231111        SZBR  OR,PUTAB,SB,LAD       CONT WRT     ;   LAD,SB:= PUTAB, WRT;
04 C2431 4655730000500100        XZ    OR,CIOST              CONT TSTBE 1 ;   TESTBE;
05 C2432 7431000000002433        JMP   F,F,.+3               NEXT         ;   INCREASE PC;
06 C2433 4645501400001003        RAR   ADD,ICS,.2,ICD                     ;   ICD:= ICS+2;
07 C2434 7405230000600000        ZQ    OR                    NEXT TSTIN   ;   NEXT INST;
08                                                                        ;  END;
09                                                                        ; END;
10 C2435 4251712000001404 AP3:   CZR   SUNO,.3,LCD           PUSH         ; FOR IX:= 1,2,3,0 DO
11                                                                        ; BEGIN
12 C2436 4644101732210011        ABQR  ADD,WRK4,WRK0,LAD     CONT READ    ;  LAD:= PUSTART+2*IX, READ;
13 C2437 0607230026000000        ZQB   OR,SB                              ;  SB:= LAD;
14 C2440 4655730000504000        XZ    OR,DATI               CONT TSTBE 1 ;  IF DATI=0 THEN   , TESTBE
15 C2441 4631000000022424        JMP   F,NZ,AP2                           ;  BEGIN
16                                                                        ;   DATO:= W;
17                                                                        ;   LAD:= PUSTART+IX+2, WRT;
18                                                                        ;   DATO:= FREE-1;
19                                                                        ;   SB,LAD:= PUTAB, WRT;
20                                                                        ;   TESTBE;
21                                                                        ;   ICD:= ICS+2, START PREFETCH;
22                                                                        ;   NEXT INST;
23 C2442 0607101432000000        ABB   ADD,.2,WRK0                        ;  END;
24 C2443 2637141772114060        NSABB AND,WRK5,WRK0,NL,F,LCN LRTN         ; END;
25 C2444 7405230000600000 APEX:  ZQ    OR                    NEXT TSTIN   ; NEXT INST, TESTINT; NO FREE PU
```

```
10074 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;DEACTIVATE PROCESS, DP, OPCODE = 72, PRIVILEGED INSTRUCTION
03
04 C2445 0624730000165640 DPE:    IZQ   OR,0,7272                                ;DPE:
05 C2446 0631000002000031         CALL  F,F,ESCAP                                ; CALL ESCAPE(7272);
06                          DP:                                                  ;DP:
07 C2447 0631000000160032         JMP   F,MMODE,INTO                             ; IF -,MONMODE THEN GOTO INTO;
08 C2450 0611541200001700         CA    AND,.1B18,STAT                           ; IF STAT(18)<>0 THEN
09 C2451 4631000000422467         JMP   T,NZ,DPEX                                ;   GOTO DP_EXIT; <* NO PROCESS ACTIVE *>
10 C2452 4603730026211111         SZBR  OR,PUTAB,SB,LAD       CONT READ          ; SB,LAD:= PUTAB, READ;
11 C2453 0654730000504000         XZQ   OR,DATI               CONT TSTBE 1       ; Q:= DATI, TESTBE;
12 C2454 0645202000000012         ZQR   ADDO,DATO                                ; DATO:= Q+1; <*NOT IN THE SAME INST*>
13 C2455 0603730026231111         SZBR  OR,PUTAB,SB,LAD       CONT WRT           ; SB,LAD:= PUTAB, WRT;
14 C2456 4643501432001100         SAB   ADD,PUTAB,.2,WRK0                        ; WRK0:= PUTAB + 2;
15 C2457 4655730000500100         XZ    OR,CIOST              CONT TSTBE 1       ; TESTBE;
16 C2460 4605240000000012         ZQR   AND,DATO                                 ; DATO:= 0;
17 C2461 0643501526231011         SABR  ADD,PUIX,WRK0,SB,LAD  CONT WRT           ; SB,LAD:= Q + PUIX, WRT;
18 C2462 0613531224001700         CAB   OR,.1B18,STAT,STAT                       ; STAT(18):= 1;<*NO PROCESS ACTIVE*>
19 C2463 4655730000500100         XZ    OR,CIOST              CONT TSTBE 1       ; TESTBE;
20 C2464 0641702000001012         SZR   ADDO,PUIX,DATO                           ; DATO:= PUIX+1;
21 C2465 4653501366233011         CABR  ADD,.6,INF,SB,LAD     CONT WRT           ; LAD,SB:= INF+6, WRT;
22 C2466 3455730000710100         XZ    OR,CIOST              NEXT TIOIN 1       ; NEXT INST, TESTIO, TESTINT;
23
24                          ; THE MONITOR LOCK IS CHANGED AS THE PROCESS IS NO LONGER ACTIVE
25
26 C2467 7405230000600000 DPEX:   ZQ    OR                    NEXT TSTIN         ; NEXT INST, TEST INT;
```

```
10075 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;EXIT, XI, OPCODE = 14, PRIVILEGED INSTRUCTION
03                          ;SB SIGNALS NEXT STATE:          0 = MONITOR LEVEL
04                          ;                                1 = USER LEVEL
05
06                          XIE:                                                ;XIE:
07 C2470 0624730000031640           IZQ   OR,U,1472                           ;
08 C2471 0631000002000031           CALL  F,F,ESCAP                           ; CALL ESCAPE(1472);
09                          XI:                                                 ;XI:
10 C2472 0631000000160032           JMP   F,MMODE,INTO                        ; IF -,MONITOR MODE THEN GOTO INTO;
11 C2473 0651523300003700           CA    SUBO,.1,SB                          ; IF SB <> 1 THEN
12 C2474 0631000000022514           JMP   F,NZ,ULEVL                          ;
13
14                          MLEVL:                                              ;MONITOR CASE:
15 C2475 4651501340211011           CAR   ADD,.14,INF,LAD       CONT READ     ; LAD:= INF+14, READ;
16 C2476 0655730000504100           XZ    OR,IOSTA              CONT TSTBE 1  ; TSTBE; <* DUMP ADDRESS *>
17 C2477 0617730026214011           XZBR  OR,DATI,SB,LAD        CONT READ     ; LAD,SB:= DATI, READ;
18 C2500 4631000002002554           CALL  F,F,LDREG                           ; LOAD REGISTERS;
19 C2501 4605240000000012           ZQR   AND,DATO                            ; DATO:= U;
20 C2502 4653501366233011           CABR  ADD,.6,INF,SB,LAD     CONT WRT      ; SB,LAD:= INF+6, WRT;
21 C2503 4624731777601360           IZQ   OR,7777,0057                        ;
22 C2504 0607041524000000           AGB   AND,WRKU,STAT                       ; STAT:= Q AND -,(12..17,19);
23 C2505 4655730000500100           XZ    OR,CIOST              CONT TSTBE 1  ; TSTBE; OPEN MONITOR LOCK
24 C2506 0605431540240011           ZAR   OR,WRK1,LAD           CONT READJ    ; LAD:= WRK1, READ INSTRUCTION;
25 C2507 4607431626000000           ZAB   OR,WRK2,SB                          ; SB:= WRK2;
26 C2510 4653501367001100           CAB   ADD,.12,INF,INF                     ; INF:= INF+12;
27 C2511 0631000000130032           JMP   F,NWADR,INTO                        ; IF W_ADDRESS THEN GOTO INTO
28 C2512 0605431540000003           ZAR   OR,WRK1,ICD                         ; ICD:= WRK1;
29 C2513 7455730000610100           XZ    OR,CIOST              NEXT TLIIN 1  ; NEXT INSTRUCTION
```

```
10076 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ULEVL:                                                  ;USER CASE:
03 C2514 0611541200CC1700           CA     AND,.1B18,STAT                           ; IF STAT(18)=C THEN
04 C2515 4631000000422547           JMP    T,NZ,ULEV2                               ; BEGIN <* PROCESS ACTIVE *>
05 C2516 4600730000CC1300           SZQ    OR,DMOFS                                 ;  Q:= DUMPOFFSET;
06 C2517 0643600026210711           SQBR   ADD,CUR,SB,LAD          CONT READ      ;  LAD,SB:= CUR+Q, READ;
07 C2520 4631000002CC2554           CALL   F,F,LDREG                                ;  LOAD REGISTERS;
08 C2521 0611541500CC2000           CA     AND,.1B17,WRK0                           ;  IF WRK0<>1B17 THEN
09 C2522 4631000000422536           JMP    T,NZ,ULEV1                               ;  BEGIN <* OPEN MONITOR LOCK *>
10 C2523 4605240000CC0012           ZQR    AND,DATO                                 ;    DATO:= 0;
11 C2524 4653501366233011           CABR   ADD,.6,INF,SB,LAD       CONT WRT       ;    SB,LAD:= INF+6, WRT;
12 C2525 0613551524CC0100           CAB    CAND,B1219,WRK0,STAT                     ;    STAT:= WRK0 AND -,(12..19);
13 C2526 0613531224CC2100           CAB    OR,.1B16,STAT,STAT                       ;    STAT:= STAT OR 1B16;
14 C2527 4653501367CC1100           CAB    ADD,.12,INF,INF                          ;    INF:= INF + 12;
15 C2530 4655730000C0100            XZ     OR,CIOST                 CONT TSTBE 1    ;    TSTBE; OPEN MONITOR LOCK
16 C2531 0605431540240011           ZAR    OR,WRK1,LAD              CONT READJ      ;    LAD:= WRK1, READ INSTRUCTION;
17 C2532 4607431626CC0000           ZAB    OR,WRK2,SB                               ;    SB:= WRK2;
18 C2533 0631000000130032           JMP    F,NWADR,INTO                             ;    IF W-ADDRESS THEN GOTO INTO;
19 C2534 0605431540CC0003           ZAR    OR,WRK1,ICD                              ;    ICD:= WRK1;
20 C2535 7455730000610100           XZ     OR,CIOST                 NEXT TLIIN 1    ;    NEXT INST, TEST LIMITS;
```

```
10077 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ULEV1:                                                  ;  END ELSE
03                                                                                  ;  BEGIN
04 02536 0605431540240011          ZAR    OR,WRK1,LAD              CONT READJ     ;   LAD:= WRK1, READ INSTRUCTION
05 02537 0611531500003605          CAR    OR,,1B20,WRK0,CPUST                     ;   CPUST:= WRK0 OR 1B20; SET DISABLE LIMI
06 02540 0613551524000100          CAB    CAND,B1219,WRK0,STAT                    ;   STAT:= WRK0 AND -,(12..19);
07 02541 0613531224002100          CAB    OR,,1B16,STAT,STAT                      ;   STAT:= STAT OR 1B16;
08 02542 4607431626000000          ZAB    OR,WRK2,SB                              ;   SB:= WRK2;
09 02543 4653501367001100          CAB    ADD,,12,INF,INF                         ;   INF:= INF+12;
10 02544 0631000000130032          JMP    F,NWADR,INTO                            ;   IF W_ADDRESS THEN GOTO INTO
11 02545 0605431540000003          ZAR    OR,WRK1,ICD                             ;   ICD:= WRK1;
12 02546 7455730000610100          XZ     OR,CIOST                 NEXT TLIIN 1   ;   NEXT INSTRUCTION
13                                                                                  ;  END;
14                          ULEV2:                                                  ; END ELSE BEGIN
15                          ;<* NO PROCESS ACTIVE, OPEN MONITOR LOCK *>
16 02547 4605240000000012          ZQR    AND,DATO                                ;  DATO:= 0;
17 02550 4653501366233011          CABR   ADD,,6,INF,SB,LAD        CONT WRT       ;  SB,LAD:= INF+6, WRT;
18 02551 4655730000500100          XZ     OR,CIOST                 CONT TSTBE 1   ;  TEST BUSERROR;
19 02552 4653501367001100          CAB    ADD,,12,INF,INF                         ;  INF:= INF+12;
20 02553 0631000000002001          JMP    F,F,DISPA                               ;  GOTO DISPATCHING POINT;
21                                                                                  ; END;
```

```
10078 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          LDREG:                                          ;LOAD REGISTERS
03 C2554 4657730020SC4000          XZB   OR,DATI,W0          CONT TSTBE 1 ; W0:= DATI;
04 C2555 4647101426210011          ABBR  ADD,.2,SB,LAD       CONT READ    ; SB,LAD:= SB+2, READ;
05 C2556 0657730021SC4000          XZB   OR,DATI,W1          CONT TSTBE 1 ; W1:= DATI; TSTBE;
06 C2557 4647101426210011          ABBR  ADD,.2,SB,LAD       CONT READ    ; SB,LAD:= SB+2, READ;
07 C2560 0657730022SC4000          XZB   OR,DATI,W2          CONT TSTBE 1 ; W2:= DATI, TSTBE;
08 C2561 4647101426210011          ABBR  ADD,.2,SB,LAD       CONT READ    ; SB,LAD:= SB+2, READ;
09 C2562 4657730023SC4000          XZB   OR,DATI,W3          CONT TSTBE 1 ; W3:= DATI, TSTBE;
10 C2563 4647101426210011          ABBR  ADD,.2,SB,LAD       CONT READ    ; SB,LAD:= SB+2, READ;
11 C2564 4657730032SC4005          XZBR  OR,DATI,WRK0,CPUST  CONT TSTBE 1 ; WRK0,CPUST:= DATI, TSTBE; LATER STAT
12 C2565 4647101426210011          ABBR  ADD,.2,SB,LAD       CONT READ    ; SB,LAD:= SB+2, READ;
13 C2566 0657730033SC4000          XZB   OR,DATI,WRK1        CONT TSTBE 1 ; WRK1:= DATI, TSTBE; LATER IC;
14 C2567 4647101426210011          ABBR  ADD,.2,SB,LAD       CONT READ    ; SB,LAD:= SB+2, READ;
15 C2570 4657730025SC4000          XZB   OR,DATI,CAUSE       CONT TSTBE 1 ; CAUSE:= DATI, TSTBE;
16 C2571 4647101426210011          ABBR  ADD,.2,SB,LAD       CONT READ    ; SB,LAD:= SB+2, READ;
17 C2572 4657730034SC4000          XZB   OR,DATI,WRK2        CONT TSTBE 1 ; WRK2:= DATI, TSTBE; LATER SB;
18 C2573 4647101426210011          ABBR  ADD,.2,SB,LAD       CONT READ    ; SB,LAD:= SB+2, READ;
19 C2574 4654730000SC4016          XZQR  OR,DATI,CPA         CONT TSTBE 1 ; CPA,Q:= DATI, TSTBE;
20 C2575 4647101426210011          AEBR  ADD,.2,SB,LAD       CONT READ    ; SB,LAD:= SB+2, READ;
21 C2576 0641220000000002          ZQS   SUB,CPAC                         ; CPAC:= Q;
22 C2577 4657730035SC4013          XZBR  OR,DATI,WRK3,BASE   CONT TSTBE 1 ; WRK3,BASE:= DATI, TSTBE;
23 C2600 4647101426210011          ABBR  ADD,.2,SB,LAD       CONT READ    ; SB,LAD:= SB+2, READ;
24 C2601 4641421640000102          ZAS   SUB,WRK3,BASEC                   ; BASEC:= WRK3;
25 C2602 0654730000SC4000          XZQ   OR,DATI             CONT TSTBE 1 ; Q:= DATI, TSTBE;
26 C2603 4647101426210011          ABBR  ADD,.2,SB,LAD       CONT READ    ; SB,LAD:= SB+2, READ;
27 C2604 4641220000000202          ZQS   SUB,LLIMC                        ; LLIMC:= Q;
28 C2605 0645013640000014          AQR   SUNO,WRK3,LLIM                   ; LLIM:= Q - BASE;
29 C2606 0654730000SC4000          XZQ   OR,DATI             CONT TSTBE 1 ; Q:= DATI, TSTBE;
30 C2607 4647101426210011          ABBR  ADD,.2,SB,LAD       CONT READ    ; SB,LAD:= SB+2, READ;
31 C2610 0641220000000302          ZQS   SUB,ULIMC                        ; ULIMC:= Q;
32 C2611 4645013640000015          AQR   SUNO,WRK3,ULIM                   ; ULIM:= Q - BASE;
33 C2612 0654730000S04021          XZQR  OR,DATI,ILIM        CONT TSTBE 1 ; Q,ILIM:= DATI, TSTBE;
34 C2613 2041220000000402          ZQS   SUB,ILIMC           RTN          ; ILIMC:= Q, RETURN;
```

```
10079 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;CHCLK:
03                          .HCLK:                                                  ;CHECK CLOCK INTERRUPT:
04 C2614 0605230000C00022          ZQR   OR,ILEVR                                   ; ILEVR:= Q;
05 C2615 4705230000C00000          ZQ    OR                        CONT 0 2         ; DELAY(675NSEC);
06 C2616 4705230000C00000          ZQ    OR                        CONT 0 2         ;
07 C2617 4724730000C07760          IZQ   OR,0,377                  CONT 2           ;
08 C2620 0617640033000300          XQB   AND,ILEV,WRK1                              ; WRK1:= ILEV(16:23);
09 C2621 4605431540C00020          ZAR   OR,WRK1,INTRG                              ; INTRG:= WRK1; CLEAR INTERRUPT
10 C2622 0745230000C00000          ZQ    OR                        CONT 0 3         ;
11 C2623 0745230000C00000          ZQ    OR                        CONT 0 3         ; DELAY(900NSEC);
12 C2624 0745230000C00000          ZQ    OR                        CONT 0 3         ;
13 C2625 0745230000C00000          ZQ    OR                        CONT 0 3         ;
14 C2626 0651523540C03600          CA    SUB0,.8,WRK1                               ; IF LEVEL=8 THEN
15 C2627 6031000000022630          JMP   F,NZ,PUINIT               RTN              ;   GOTO PUINIT
16                                                                                  ; ELSE RETURN;
```

```
1008C CPU823 MICROPROGRAM LBJ 850816 VERSION 8

                                PUINIT:                                                         ;
   C2630 46137300370C3400             CZB   OR,.7,WRK5                                          ; MASK:= 7;
   C2631 061173000021361 1            CZR   OR,.8,LAD                 CONT READ                 ; LAD:= 8, READ;
   C2632 4655541740004100             XA    AND,IOSTA,WRK5            CONT 0 1                  ; IF IOSTA AND MASK <> 0 THEN
   C2633 4631000000423004             JMP   T,NZ,SYFL7                                          ;  GOTO SYSTEMFAULT7;
   C2634 462473007600000 0            IZQ   OR,0370,0                                           ;
   C2635 0617640032CC0100             XGB   AND,CIOST,WRK0                                      ; WRK0:= CIOST AND 8'3700000;
   C2636 4614730000CC4000             XZQ   OR,DATI                                             ; Q:= DATI;
   C2637 0722601532CC0600             HGB   ADD,LSWAP,WRK0,WRK0                                 ; WRK0:= Q + WRK0(0:11);
   C2640 4605431500210011             ZAR   OR,WRK0,LAD               CONT READ                 ; LAD:= WRK0, READ;
   C2641 4655541740004100             XA    AND,IOSTA,WRK5            CONT 0 1                  ; IF IOSTA AND MASK <> 0 THEN
   C2642 4631000000423004             JMP   T,NZ,SYFL7                                          ;  GOTO SYSTEMFAULT7;
   C2643 4617730033004000             XZB   OR,DATI,WRK1                                        ; WRK1:= DATI;
   C2644 0605431540210011             ZAR   OR,WRK1,LAD               CONT READ                 ; LAD:= WRK1, READ;
   C2645 4655541740004100             XA    AND,IOSTA,WRK5            CONT 0 1                  ; IF IOSTA AND MASK <> 0 THEN
   C2646 4631000000423004             JMP   T,NZ,SYFL7                                          ;  GOTO SYSTEMFAULT7;
   C2647 0617730034004000             XZB   OR,DATI,WRK2                                        ; WRK2:= DATI;
   C2650 4641421600001102             ZAS   SUB,WRK2,PUTAB                                      ; PUTAB:= WRK2;
   C2651 0647101433210011             ABBR  ADD,.2,WRK1,LAD           CONT READ                 ; WRK1,LAD:= WRK1+2, READ;
   C2652 4655541740004100             XA    AND,IOSTA,WRK5            CONT 0 1                  ; IF IOSTA AND MASK <> 0 THEN
   C2653 4631000000423004             JMP   T,NZ,SYFL7                                          ;  GOTO SYSTEMFAULT7;
   C2654 4617730027004000             XZB   OR,DATI,INF                                         ; INF:= DATI;
   C2655 0647101433210011             ABBR  ADD,.2,WRK1,LAD           CONT READ                 ; WRK1,LAD:= WRK1+2, READ;
   C2656 4655541740004100             XA    AND,IOSTA,WRK5            CONT 0 1                  ; IF IOSTA AND MASK <> 0 THEN
   C2657 4631000000423004             JMP   T,NZ,SYFL7                                          ;  GOTO SYSTEMFAULT7;
   C2660 4614730000004000             XZQ   OR,DATI                                             ;
   C2661 0641220000000602             ZQS   SUB,MONT                                            ; MONT:= DATI;
   C2662 0647101433210011             ABBR  ADD,.2,WRK1,LAD           CONT READ                 ; WRK1,LAD:= WRK1+2, READ;
   C2663 4655541740004100             XA    AND,IOSTA,WRK5            CONT 0 1                  ; IF IOSTA AND MASK <> 0 THEN
   C2664 4631000000423004             JMP   T,NZ,SYFL7                                          ;  GOTO SYSTEMFAULT7;
   C2665 4614730000004000             XZQ   OR,DATI                                             ;
   C2666 0641220000000502             ZQS   SUB,SIZE                                            ; SIZE:= DATI;
   C2667 0647101433210011             ABBR  ADD,.2,WRK1,LAD           CONT READ                 ; WRK1,LAD:= WRK1+2, READ
   C2670 4655541740004100             XA    AND,IOSTA,WRK5            CONT 0 1                  ; IF IOSTA AND MASK <> 0 THEN
```

```
10081 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 C2671 4631000000423004          JMP   T,NZ,SYFL7                          ;   GOTO SYSTEMFAULT7;
03 C2672 4614730000C04000          XZQ   OR,DATI                             ;
04 C2673 0641220000C01202          ZQS   SUB,EXOFS                           ; EXOFS:= DATI;
05 C2674 0647101433210011          ABBR  ADD,.2,WRK1,LAD       CONT READ     ; WRK1,LAD:= WRK1+2, READ
06 C2675 4655541740C04100          XA    AND,IOSTA,WRK5        CONT 0 1      ; IF IOSTA AND MASK <> 0 THEN
07 C2676 4631000000423004          JMP   T,NZ,SYFL7                          ;   GOTO SYSTEMFAULT7;
08 C2677 4614730000C04000          XZQ   OR,DATI                             ;
09 C2700 4641220000C01302          ZQS   SUB,DMOFS                           ; DMOFS:= DATI;
10 C2701 0611730000C01424          CZR   OR,.3,CAMBC                         ; CAMBC:= 3; SET BYPASSCAM+BUSBUSY
11 C2702 0651712000C01404          CZR   SUNO,.3,LCD                         ;
12 C2703 4607240032C00000          ZQB   AND,WRK0                            ; WRK0:= 0;
13 C2704 4207101434C00000          ABB   ADD,.2,WRK2           PUSH          ; WRK2:= WRK2+2;
14                                                                           ; FOR LC:= -3 STEP 1 UNTIL 0 D
15                                                                           ; BEGIN
16 C2705 4645101534210011          ABR   ADD,WRK0,WRK2,LAD     CONT READ     ;  LAD:= WRK2+WRK0, READ;
17 C2706 4655541740C04100          XA    AND,IOSTA,WRK5        CONT 0 1      ;  IF IOSTA AND MASK <> 0 THEN
18 C2707 0631000000422777          JMP   T,NZ,ERR                            ;   GOTO ERROR;
19 C2710 0655702000C04000          XZ    ADDO,DATI                           ;  IF DATI=-1 THEN
20 C2711 4631000000022714          JMP   F,NZ,FOUND                          ;   GOTO FOUND;
21 C2712 2637101434114060          NSABB ADD,.2,WRK0,NL,F,LCN  LRTN          ;  WRK0:= WRK0+2;
22                                                                           ; END;
23 C2713 0631000000C02777          JMP   F,F,ERR                             ; GOTO ERROR;
24                          FOUND:                                           ; FOUND:
25 C2714 0641421500C01002          ZAS   SUB,WRK0,PUIX                       ; PUIX:= WRK0;
26 C2715 4613531471CC2006          CABR  OR,.1B17,CNTR,CNTR,CNTR0            ; CNTR,CNTR0:= CNTR OR 1B17;
27                                                                           ; TURN RUNLAMP ON OCP ON;
```

```
10082 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 C2716 4603730032211111          SZBR  OR,PUTAB,WRK0,LAD      CONT READ    ; LAD,WRK0:= PUTAB, READ;
03 C2717 4655541740004100          XA    AND,IOSTA,WRK5         CONT U 1     ; IF IOSTA AND MASK <> 0 THEN
04 C2720 0631000000422777          JMP   T,NZ,ERR                            ;  GOTO ERROR;
05 C2721 4614730000004000          XZQ   OR,DATI                             ; Q:= DATI;
06 C2722 0645202000000012          ZQR   ADD0,DATO                           ; DATO:= Q+1;
07 C2723 0605431500230011          ZAR   OR,WRK0,LAD            CONT WRT     ; LAD:= WRK0, WRT;
08 C2724 0607101432000000          ABB   ADD,.2,WRK0                         ; WRK0:= WRK0 + 2;
09 C2725 0655541740000100          XA    AND,CIOST,WRK5         CONT U 1     ; IF CIOST AND MASK <> 0 THEN
10 C2726 0631000000422777          JMP   T,NZ,ERR                            ;  GOTO ERROR;
11 C2727 4611730000003724          CZR   OR,.1,CAMBC                         ; CAMBC:= 1;
12 C2730 4605240000000012          ZQR   AND,DATO                            ; DATO:= 0;
13 C2731 0641501500231011          SAR   ADD,PUIX,WRK0,LAD      CONT WRT     ; LAD:= WRK0 + PUIX, WRT;
14 C2732 0655541740000100          XA    AND,CIOST,WRK5         CONT U 1     ; IF CIOST AND MASK <> 0 THEN
15 C2733 4631000000423002          JMP   T,NZ,ERR1                           ;  GOTO ERROR1;
16 C2734 4605240000000024          ZQR   AND,CAMBC                           ; CAMBC:= 0;
17 C2735 4653501367001100          CAB   ADD,.12,INF,INF                     ; INF:= INF+12;
18                            .ISPA:                                        ;DISPATCHING POINT:
19 C2736 0624730017601760          IZQ   OR,63.,63.                          ;
20 C2737 0605230000000021          ZQR   OR,ILIM                             ; ILIM:= 63<12+63; ENABLE ALL INT
21 C2740 4643501432001100          SAB   ADD,PUTAB,.2,WRK0                   ; WRK0:= PUTAB + 2;
22 C2741 4613531471001506          CABR  OR,.4,CNTR,CNTR,CNTR0               ; CNTR,CNTR0:= CNTR OR 4; CLEAR RUN LAMP
23 C2742 0624730000005000          IZQ   OR,U,0240                           ;
24 C2743 4607230024000000          ZQB   OR,STAT                             ; STAT:= 1B16+1B18;
25 C2744 4600730000001000          SZQ   OR,PUIX                             ;
26 C2745 4601230000000702          ZQS   OR,CUR                              ; CUR:= -PUIX-1;
27
28                            ; WHILE IDLE, THE PU IS ON USER LEVEL
29
30 C2746 4643501532211011          SABR  ADD,PUIX,WRK0,WRK0,LAD CONT READ    ; WRK0,LAD:= WRK0+PUIX, READ;
31 C2747 0657730035004000          XZB   OR,DATI,WRK3           CONT U 1     ; WRK3:= DATI, WAIT;
32 C2750 4631000000422754          JMP   T,NZ,DISP1                          ; IF WRK3 = 0 THEN
```

```
10083 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ; DO NOT TEST FOR INTERRUPT THE FIRST TIME (SINGLE STEP)
03
04                          LOP1:                                                   ; REPEAT
05 C2751 4605431500210011           ZAR   OR,WRK0,LAD             CONT READ     ;  LAD:= WRK0, READ;
06 C2752 0657730035604000           XZB   OR,DATI,WRK3            CONT ISTIN 1  ;  WRK3:= DATI, ISTINT;
07 C2753 0631000000022751           JMP   F,NZ,LOP1                             ; UNTIL WRK3 <> 0;
08 C2754 4641421640CC0702 DISP1:    ZAS   SUB,WRK3,CUR                          ; CUR:= WRK3;
09 C2755 0643501666211311           SABR  ADD,DMOFS,WRK3,SB,LAD CONT READ       ; LAD,SB:= WRK3+DMOFS, READ;
10
11                          ; WHILE LOADING THE REGISTERS, THE PU IS ON MONITOR LEVEL
12
13 C2756 4653513367001100          CAB   SUN0,.12,INF,INF                      ; INF:= INF - 12;
14 C2757 0607240024CC0000          ZGB   AND,STAT                              ; STAT:= 0;
15 C2760 4613551471CC1506          CABR  CAND,.4,CNTR,CNTR,CNTR0               ; CNTR,CNTR0:= CNTR CAND 4; SET RUN LAMP
16 C2761 4631000002CC2554          CALL  F,F,LDREG                             ; LOAD REGISTERS;
17 C2762 4653501367001100          CAB   ADD,.12,INF,INF                       ; INF:= INF + 12;
18                                                                             ; CHANGE TO USER LEVEL;
19 C2763 0613551524CC0100          CAB   CAND,B1219,WRK0,STAT                  ; STAT:= WRK0 AND -,(12..19);
20 C2764 0611541500CC2000          CA    AND,.1B17,WRK0                        ; IF LOCK CONDITION THEN
```

```
!0084 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 C2765 4631000000022771        JMP   F,NZ,DISP2                          ; BEGIN
03 C2766 0631000002C00037        CALL  F,F,LOCKM                           ;  LOCKMONITOR;
04 C2767 0601730000CCC0421        SZR   OR,ILIMC,ILIM                       ;  RESTORE INTERRUPTLIMITS;
05 C2770 4611531200CC3605        CAR   OR,.1B20,STAT,CPUST                 ;  SET DISABLE LIMITS;
06                         DISP2:                                          ; END;
07 C2771 0605431540240011        ZAR   OR,WRK1,LAD          CONT READJ     ; LAD:= WRK1, READ INSTRUCTION;
08 C2772 4607431626CC0000        ZAB   OR,WRK2,SB                          ; SB:= WRK2;
09 C2773 0613531224CC2100        CAB   OR,.1B16,STAT,STAT                  ; STAT:= STAT OR 1B16;
10 C2774 0605431540CC0003        ZAR   OR,WRK1,ICD                         ; ICD:= WRK1;
11 C2775 0631000000130032        JMP   F,NWADR,INTO                        ; IF W_ADDRESS THEN GOTO INTC
12 C2776 7455730000610100        XZ    OR,CIOST             NEXT TLIIN 1 ; GOTO NEXT INSTRUCTION;
13
14                         ERR:                                            ;ERROR:
15 C2777 4611730000003724        CZR   OR,.1,CAMBC                         ; CAMBC:= 1; SET BYPASSCAM
16 C3000 4605240000210011        ZQR   AND,LAD              CONT READ      ; LAD:= 0, READ;
17 C3001 4655730000000100        XZ    OR,CIOST             CONT 0 1       ; SYNC IO;
18                         ERR1:                                           ;ERROR1:
19 C3002 4605240000CC0024        ZQR   AND,CAMBC                           ; CAMBC:= 0;
20 C3003 4631000000CC3004        JMP   F,F,SYFL7                           ; GOTO SYSTEMFAULT7;
21
22                         SYFL7:                                          ;SYSTEMFAULT7:
23 C3004 4624730000000160        IZQ   OR,0,7                              ; ERROR IN INITIALIZING PU;
24 C3005 0631000002CC4021        CALL  F,F,SYFLT                           ; SYSTEMFAULT(7);
```

```
10085 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;MOVE HALFWORDS, MH, OPCODE = 73
03
04                          MHE:                                                ;MHE:
05 C3006 0624730000166340           IZQ    OR,0,7316                        ;
06 C3007 0631000002000031           CALL   F,F,ESCAPE                       ; CALL ESCAPE(7316);
07
08                          MH:                                                 ;MH:
09 C3010 4645523400CC1003           RAR    SUB0,ICS,,2,ICD                  ; ICD:= ICS -  2;
10 C3011 4613531224CC3105           CABR   OR,,1B3,STAT,STAT,CPUST          ; STAT,CPUST:= STAT OR 1B3;
11                          ; <* SET AFTER ESCAPE IN ORDER TO INHIBIT ADDRESS CALCULATION *>
12 C3012 4613551326CC3700           CAB    CAND,,1,SB,SB                    ; SB:= SB AND -,1;<*MAKE SB EVEN*>
13 C3013 0631000000023037           JMP    F,NZ,MHFIN                       ; IF SB=0 THEN GOTO MHFIN;
14 C3014 4605122001CC0000           AB     SUB0,W,WPRE                      ; IF W >= WPRE THEN
15 C3015 4231000000413030           JMP    T,NEG,MHBAK          PUSH        ; BEGIN <* MOVE FORWARD *>
16 C3016 4655730000710100           XZ     OR,CIOST             CONT TIOIN 1 ;  FOR SB:= SB TO 0 BY -2 DO
17                                                                          ;  BEGIN
18                                                                          ;   TESTIO, TESTINTERRUPT;
19 C3017 4605430000200011           ZAR    OR,W,LAD             CONT READP   ;   LAD:= W, READP;
20 C3020 0655730000514012           XZR    OR,DATI,DATO         CONT TSTIO 1 ;   DATO:= DATI, TESTIO;
21 C3021 4605430040220011           ZAR    OR,WPRE,LAD          CONT WRTP    ;   LAD:= WPRE, WRTP;
22 C3022 4607101400CC0000           ABB    ADD,,2,W                         ;   W:= W+2;
23 C3023 0607113426CC0000           ABB    SUN0,,2,SB                       ;   WPRE:= WPRE+2;
24 C3024 6637101401C24060           NSABB  ADD,,2,WPRE,NL,F,NZ  LRTN        ;  END;
25 C3025 4613551224CC3105           CABR   CAND,,1B3,STAT,STAT,CPUST        ;  STAT,CPUST:= STAT OR 1B3;
26 C3026 4645501400CC1003           RAR    ADD,ICS,,2,ICD                   ;  ICD:= ICS+2;
27 C3027 3455730000710100           XZ     OR,CIOST             NEXT TIOIN 1 ;  TESTIO,TESTINT, NEXT INST;
```

```
10086 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          MHBAK:                                                ; END FORWARD;
03                                                                                ; ELSE BEGIN
04 C3030 4253511333OC3700          CAB    SUN,.1,SB,WRK1          PUSH          ;  I:= SB-2;
05                                                                                ;  FOR J:=I TO 0 BY -2 DO
06                                                                                ;  BEGIN
07 C3031 4655730000710100          XZ     OR,CIOST                CONT TIOIN 1  ;    TESTIO, TESTINT;
08 C3032 0645101540200011          ABR    ADD,WRK1,W,LAD          CONT READP    ;    LAD:= W+I, READP;
09 C3033 0655730000514012          XZR    OR,DATI,DATO            CONT TSTIO 1  ;    DATO:= DATI, TESTIO;
10 C3034 0645101541220011          ABR    ADD,WRK1,WPRE,LAD       CONT WRTP     ;    LAD:= WPRE+I, WRTP;
11 C3035 0607113426CC0000          ABB    SUN0,.2,SB                            ;    SB:= SB-2;
12 C3036 2637113433C24060          NSABB  SUN0,.2,WRK1,NL,F,NZ    LRTN          ;  END;
13                          MHFIN:                                                ; END BACKWARD;
14 C3037 4613551224CC3105          CABR   CAND,.1B3,STAT,STAT,CPUST               ; STAT,CPUST:= STAT AND -,1B3;
15 C3040 4645501400CC1003          RAR    ADD,ICS,.2,ICD                         ; IC:= IC+2;
16 C3041 3455730000710100          XZ     OR,CIOST                NEXT TIOIN 1  ; TESTIO TESTINT;
```

```
10087 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;INDEX CHECK AND CALCULATION, IX, OPCODE = 75
03                          ;THE INSTRUCTION HAS THE FOLLOWING FORMAT
04                          ; IX W <BASE>
05                          ; <DOPEREL>,<TYPE>
06
07                          IXE:                                                ;IXE:
08 C3042 0624730000172640           IZQ   OR,0,7532                           ;
09 C3043 0631000002000031           CALL  F,F,ESCAP                           ; CALL ESCAPE(7532);
10                          IX:                                                 ;IX:
11 C3044 4645501400C01003           RAR   ADD,ICS,.2,ICD                      ; ICD:= ICS+2;
12 C3045 0631000000130032           JMP   F,NWADR,INTO                        ; IF SB < 8 THEN GOTO INTO;
13 C3046 4654730000514000           XZQ   OR,DATI                CONT TSTIO 1 ; Q:= DATI, TESTIO;
14 C3047 0645523400201011           RAR   SUBO,ICS,.2,LAD        CONT READP   ; LAD:= ICS-2, READP; READ DOPEREL,TYPE
15 C3050 0607430037CC0000           ZAB   OR,W,WRK5                           ; IX:= W;
16 C3051 4657730033514000           XZB   OR,DATI,WRK1           CONT TSTIO 1 ; WRK1:= DATI, TSTIO;
17 C3052 4662731572CC0100           HZB   OR,EXLOA,WRK1,WRKO                  ; DOPEREL:=WRK1(0)EXTEND12 CON WRK1(0:11);
18 C3053 4647101526200011           ABBR  ADD,WRKO,SB,LAD        CONT READP   ; SB,LAD:= BASE+DOPEREL, READP; (* LB *)
19 C3054 0662731573CC0400           HZB   OR,EX1,WRK1,WRK1                    ; TYPE:=WRK1(12)EXT 12 CON WRK1(12:23);
20                          ; NOTE!! HZB DOES NOT WRITE THE RESULT ON DBUS, SO NO CONDITION IS GENERATED
21 C3055 0605423540CC0000           ZA    SUBO,WRK1                           ; IF TYPE > 0 THEN
22 C3056 0631000000C13064           JMP   F,NEG,IXNEG                         ; BEGIN
23 C3057 4251523540C03704           CAR   SUBO,.1,WRK1,LCD       PUSH         ;  LCD:= 1-TYPE;
24 C3060 6637330037520060           SLZBB OR,WRK5,Z,NL,T,MAXLO   LRTN         ;  IX:= IX SHIFT TYPE;
25 C3061 4657730035514000           XZB   OR,DATI,WRK3           CONT TSTIO 1 ;  LB:= DATI, TSTIO;
26 C3062 0647113426200011           ABBR  SUNO,.2,SB,LAD         CONT READP   ;  SB,LAD:= SB-2, READP; (* UB *)
27 C3063 0631000000CC3066           JMP   F,F,IXJOIN                          ; END ELSE
```

```
10C88 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          IXNEG:                                          ; BEGIN
03 C3064 0717501575514000           XAB   ADD,DATI,WRK1,WRK3    CONT TSTIO 2 ;  LB:= DATI+TYPE, TSTIO;
04 C3065 0647113426200011           ABBR  SUNO,.2,SB,LAD        CONT READP   ;  SB,LAD:= SB-2, READP;
05                          IXJOIN:                                         ; END;
06 C3066 4657730036514000           XZB   OR,DATI,WRK4          CONT TSTIO 1 ; UB:= DATI, TSTIO;
07 C3067 4605730000241011           RZR   OR,ICS,LAD            CONT READJ   ; LAD:= ICS, READJ; (* START PREFETCH *)
08 C3070 4605123737CC0000           AB    SUBO,WRK4,WRK5                     ; IF IX <= UB AND
09 C3071 4631000000413076           JMP   T,NEG,IXEXC                        ;
10 C3072 4605123677CC0000           AB    SUBO,WRK3,WRK5                     ; IX > LB THEN
11 C3073 0631000000C13076           JMP   F,NEG,IXEXC                        ; BEGIN
12 C3074 4607001740CC0000           AQB   ADD,WRK5,W                         ;  W:= IX+WORD(BASE);
13 C3075 7413551224601400           CAB   CAND,.3,STAT,STAT     NEXT TSTIN   ;  STAT(22:23):=0, NEXT INT, TESTINT;
14                                                                           ; END ELSE
15                                                                           ; BEGIN
16 C3076 0613531224001400 IXEXC:    CAB   OR,.3,STAT,STAT                    ;  STAT(22:23):= 3;
17 C3077 4611541200CC2600           CA    AND,.1B5,STAT                      ;  IF STAT(5) <> 0 THEN(* FP ACTIVE *)
18 C3100 0631000000420035           JMP   T,NZ,INTF                          ;   GOTO INTF;
19 C3101 7407001700600000           AQB   ADD,WRK4,W            NEXT TSTIN   ;  W:=WORD(BASE)+UB; NEXT INT,TSTIN
20                                                                           ; END;
```

```
10089 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02
03                          FILL 40CC
04                          .EOT
05
06                          ;FILE: T3, EDITED: 850402/LBJ
07
08                          ;*************************************************
09                          ;
10                          ;          CPU 823 TCP ROUTINE
11                          ;
12                          ;*************************************************
13
14                  CC0001          .NOMAC' 1
15                  014000          .LOC 4000*3
16
17                          ;JUMP TABLE STARTING AT LOCATION 8'4000
18
19 C4000 4631000000004617          JMP   F,F,.OWUP                    ; GOTO POWERUP
20 C4001 4631000000004022          JMP   F,F,.CPIN                    ; GOTO TCPIN
21 C4002 4631000000005073          JMP   F,F,.CPAL                    ; GOTO OCPAUTOLOAD
22 C4003 4631000000005111          JMP   F,F,.EMAL                    ; GOTO REMOTE AUTOLOAD
23 C4004 4631000000004511          JMP   F,F,.INGL                    ; GOTO SINGLE INSTR. INTERRUPT
24 C4005 4631000000004207          JMP   F,F,.CPER                    ; GOTO TCPERROR
25 C4006 4631000000004137          JMP   F,F,.POCT                    ; GOTO TYPE OCTAL
26 C4007 4631000000004162          JMP   F,F,.CRLF                    ; GOTO TYPE 'CR''LF'
27 C4010 0631000000004075          JMP   F,F,.CHAR                    ; GOTO GETCHAR
28 C4011 4631000000004024          JMP   F,F,.CPI2                    ; GOTO TCPIN2
29 C4012 0631000000005035          JMP   F,F,.ELAY                    ; GOTO DELAY
30 C4013 4631000000004164          JMP   F,F,.PTXT                    ; GOTO TYPE TEXT
31 C4014 4631000000004116          JMP   F,F,.TNUM                    ; GOTO GETNUMBER
32 C4015 4631000000005106          JMP   F,F,.AUTO                    ; GOTO RAUTO
33 C4016 4631000000005037          JMP   F,F,.INTR                    ; GOTO CLEAR INTERRUPT
34 C4017 0631000000005366          JMP   F,F,.AMTEST                  ; GOTO CAMTEST
35 C4020 4631000000005357          JMP   F,F,.AMER                    ; GOTO CAM ERROR
36 C4021 4631000000005247          JMP   F,F,.YFLT                    ; GOTO SYFLT
```

```
1009C CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;TCPIN:                                          ;
03                          .CPIN:                                           ;
04 C4022 4641421500001402           ZAS    SUB,WRK0,SWRK0                    ; SAVE WRK0 IN SWRK0
05 C4023 4631000002CC4010           CALL   F,F,GCHAR                         ; Q:= GETCHAR
06                          .CPI2:                                           ;
07                          ;TCPI2:                                          ;
08 C4024 0611541440CC2000           CA     AND,.1B17,CNTR                    ; IF CNTR.RUN AND Q= 'BELL'
09 C4025 0631000000C24033           JMP    F,NZ,TCPI3                        ; BEGIN
10 C4026 0665622000C00160           IQ     SUB0,0,7                          ;
11 C4027 0631000000C24033           JMP    F,NZ,TCPI3                        ;
12 C4030 0611541440CC1500           CA     AND,.4,CNTR                       ;   IF CNTR.21 <> 0 THEN
13 C4031 0631000000422001           JMP    T,NZ,DISPA                        ;   GOTO DISPATCHING POINT <* NOT RUN *>
14 C4032 4631000000C04500           JMP    F,F,C1                            ;   ELSE GOTO CONTINUE;
15                                                                           ; END;
16                          TCPI3:                                           ;TCPI3:
17 C4033 0665622000CC2060           IQ     SUB0,0,"C                         ; IF Q='C' THEN GOTO C
18 C4034 0631000000C24473           JMP    F,NZ,C                            ;
19 C4035 0665622000CC2300           IQ     SUB0,0,"L                         ;
20 C4036 4631000000424047           JMP    T,NZ,TESTR                        ; IF Q<>'L' THEN GOTO TEST_R
21 C4037 4631000002CC4010           CALL   F,F,GCHAR                         ; Q:=GETCHAR
22 C4040 4665622000CC2320           IQ     SUB0,0,"M                         ; IF Q='M' THEN GOTO LM
23 C4041 4631000000024421           JMP    F,NZ,LM                           ;
24 C4042 4665622000CC2340           IQ     SUB0,0,"N                         ; IF Q='N' THEN GOTO LN
25 C4043 0631000000C24425           JMP    F,NZ,LN                           ;
26 C4044 0665622000CC2440           IQ     SUB0,0,"R                         ; IF Q='R' THEN GOTO LR
27 C4045 0631000000C24407           JMP    F,NZ,LR                           ;
28 C4046 4631000000CC4005           JMP    F,F,TCPER                         ; GOTO TCP ERROR
```

```
10091 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 C4047 0665622000C2440 TESTR: IQ    SUB0,0,"R                ; IF Q='R' THEN GOTO R
03 C4050 0631000000C24416        JMP   F,NZ,R                   ;
04 C4051 4665622000C2460         IQ    SUB0,0,"S                ; IF Q='S' THEN GOTO S
05 C4052 0631000000024464        JMP   F,NZ,S                   ;
06 C4053 0665622000C2500         IQ    SUB0,0,"T                ; IF Q='T' THEN GOTO TX
07 C4054 4631000000C26000        JMP   F,NZ,TX                  ;
08 C4055 0665622000C2600         IQ    SUB0,0,"X                ; IF Q<>'X' THEN GOTO TCP ERROR
09 C4056 4631000000424005        JMP   T,NZ,TCPER               ;
10 C4057 4631000002C04010        CALL  F,F,GCHAR                ; Q:= GETCHAR
11 C4060 4665622000C2100         IQ    SUB0,0,"D                ; IF Q='D' THEN GOTO XD
12 C4061 0631000000C24503        JMP   F,NZ,XD                  ;
13 C4062 4665622000C2320         IQ    SUB0,0,"M                ; IF Q='M' THEN GOTO XMM
14 C4063 0631000000C24440        JMP   F,NZ,XMM                 ;
15 C4064 4665622000C2340         IQ    SUB0,0,"N                ; IF Q='N' THEN GOTO XN
16 C4065 4631000000C24444        JMP   F,NZ,XN                  ;
17 C4066 0665622000C2440         IQ    SUB0,0,"R                ; IF Q='R' THEN GOTO XR
18 C4067 4631000000C24400        JMP   F,NZ,XR                  ;
19 C4070 4665622000C2460         IQ    SUB0,0,"S                ; IF Q='S' THEN GOTO XST
20 C4071 0631000000C24506        JMP   F,NZ,XST                 ;
21 C4072 0665622000C2560         IQ    SUB0,0,"W                ; IF Q='W' THEN GOTO XW
22 C4073 0631000000C24577        JMP   F,NZ,XW                  ;
23 C4074 4631000000C04005        JMP   F,F,TCPER                ; GOTO TCP ERROR
```

```
10092 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ; CHARACTER INPUT PROCEDURE
03
04                          ;THE PROCEDURE READS A 8 BITS CHARACTER FROM TCP AND MASK OUT THE 7 LEAST
05                          ;SIGNIFICANT BITS. THE RESULT IS LOADED IN THE Q-REGISTER AND THE WRK0-REGISTER
06                          ;AND ECHOED ON TCPOUT
07
08                          ; REGISTERS USED: WRK0
09
10                          .CHAR:                                                  ;
11                          ;GCHAR:                                                 ;GETCHAR:
12 C4075 4615541400CC0500           XA     AND,TCPST,.2                         ; IF TCPST(22)<>0 THEN WAIT
13 C4076 4631000000C24010           JMP    F,NZ,GCHAR                           ; C. DATA AVAILABLE
14 C4077 4614730000CC0400           XZQ    OR,TPIN                              ; Q:=TCPIN
15 C4100 4605230000CC0025           ZGR    OR,TCCLR                             ; CLEAR TC INTERRUPT
16 C4101 4624640000C03760           IQQ    AND,0,177                            ; Q:=Q(17:23)
17 C4102 0607230032CC0000           ZQB    OR,WRK0                              ; WRK0:=Q
18 C4103 4631000000C04104           JMP    F,F,TYPE                             ; TYPE(WRK0)
19
20                          ;PROCEDURE TYPE
21
22                          ;THE CHARACTER IN THE WRK0-REGISTER IS TYPED ON THE TCP
23
24                          ;REGISTER USED: Q, RETURN: Q=CHAR
25
26                          TYPE:                                                   ;TYPE:
27 C4104 4624730000CC0020           IZQ    OR,0,1                               ; Q:=1
28 C4105 4615640000CC0500 WOUT:     XQ     AND,TCPST                            ; IF TCPST(23) THEN WAIT
29 C4106 0631000000424105           JMP    T,NZ,WOUT                            ; C. TCOUTFULL
30 C4107 0624730000CC7760           IZQ    OR,0,377                             ;
31 C4110 4604041500CC0023           AQQR   AND,WRK0,TPOUT                       ; Q:=TPOUT:= WRK0(16:23)
32 C4111 0665622000CC0320           IQ     SUB0,0,15                            ; IF Q<>CR THEN RETURN
33 C4112 6031000000C24113           JMP    F,NZ,.+3                 RTN         ;
34 C4113 0624730000CC0240           IZQ    OR,0,12                              ;
35 C4114 0607230032CC0000           ZQB    OR,WRK0                              ; WRK0:= LF
36 C4115 4631000000C04104           JMP    F,F,TYPE                             ; GOTO TYPE
```

```
10093 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ; OCTAL NUMBER INPUT PROCEDURE
03                          ;
04                          ;THE PROCEDURE READS AN OCTAL NUMBER FROM TCP TO WRK2.
05                          ;THE NUMBER IS TERMINATED BY 'LF' OR ':'
06                          ; REGISTER USED: Q,WRK0
07 ;
08                          .TNUM:                                              ;
09                          ;GTNUM:                                             ;GETNUM:
10 04116 0607441634000000           ZAB   AND,WRK2,WRK2                       ; WRK2:=0
11 04117 4631000002004010 GTNU1:    CALL  F,F,GCHAR                           ; Q:= GETCHAR
12 04120 4665622000CC0240           IQ    SUB0,0,12                           ; IF Q='LF' THEN RETURN
13 04121 6031000000424122           JMP   T,NZ,.+3              RTN           ;
14 04122 4665622000CC1640           IQ    SUB0,0,":                           ; IF Q=':' THEN RETURN
15 04123 6031000000424124           JMP   T,NZ,.+3              RTN           ;
16 04124 0665622000CC1000           IQ    SUB0,0,40                           ;
17 04125 4631000000C24117           JMP   F,NZ,GTNU1                          ; IF Q='SP' THEN GOTO GTNU1
18 04126 0665622000CC1360           IQ    SUB0,0,57                           ; IF Q<= 8'57 THEN
19 04127 0631000000014005           JMP   F,NEG,TCPER                         ;   GOTO TCP ERROR
20 04130 0665622000CC1560           IQ    SUB0,0,67                           ; IF Q> 8'67 THEN
21 04131 4631000000414005           JMP   T,NEG,TCPER                         ;   GOTO TCP ERROR
22 04132 4624640000CC0160           IQQ   AND,0,7                             ; Q:= 21 EXT 0 CON Q(21:23)
23 04133 0677101634CC0060           SLABB ADD,WRK2,WRK2,Z,NL,F,F              ; WRK2:=WRK2 SHIFT 2
24 04134 4637330034CC0060           SLZBB OR,WRK2,Z,NL,F,F                    ; WRK2:=WRK2 SHIFT 1
25 04135 0607031634C00000           AQB   OR,WRK2,WRK2                        ; WRK2:=WRK2 OR Q
26 04136 0631000000C04117           JMP   F,F,GTNU1                           ; GOTO GTNU1
```

```
10094 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;TYPE OCTAL NUMBER PROCEDURE
03                          ;
04                          ;THE PROCEDURE TYPES THE CONTENT OF THE Q-REGISTER AS 8 OCTAL DIGITS
05                          ; REGISTERS USED: WRK0,WRK1,WRK2
06
07                          .POCT:                                                  ;
08                          ;TPOCT:                                                 ;TYPEOCTAL:
09 C4137 4607230033CC0000           ZQB    OR,WRK1                                 ; WRK1:= Q
10 C4140 4624730000CC0200           IZQ    OR,0,10                                 ;
11 C4141 0607230034CC0000           ZQB    OR,WRK2                                 ; WRK2:= 8 , COUNT
12                          TPOC1:                                                  ; REPEAT
13 C4142 0607340032CC0000           ZEB    AND,WRK0                                ;   WRK0:=0
14 C4143 4604431540CC0000           ZAQ    OR,WRK1                                 ;   Q:=WRK1
15 C4144 0636330032CC0060           DLZBB  OR,WRK0,Z,NL,F,F                        ;   WRK0,Q:= WRK0,Q SHIFT 1
16 C4145 0636330032CC0060           DLZBB  OR,WRK0,Z,NL,F,F                        ;   WRK0,Q:= WRK0,Q SHIFT 1
17 C4146 0636330032CC0060           DLZBB  OR,WRK0,Z,NL,F,F                        ;   WRK0,Q:= WRK0,Q SHIFT 1
18 C4147 4607230033CC0000           ZQB    OR,WRK1                                 ;   WRK1:=Q
19 C4150 0624730000CC1400           IZQ    OR,0,60                                 ;   Q:=8'60
20 C4151 0607031532CC0000           AQB    OR,WRK0,WRK0                            ;   WRK0:= WRK0 OR Q
21 C4152 0631000002CC4104           CALL   F,F,TYPE                                ;   TYPE(WRK0)
22 C4153 0607310034CC0000           ZBB    SUN,WRK2                                ;   COUNT:=COUNT-1
23 C4154 6031000000424142           JMP    T,NZ,TPOC1              RTN             ; UNTIL COUNT = 0
24                                                                                 ; RETURN
25
26                          ;TYPE ADDRESS
27
28                          ;THE PROCEDURE TYPES THE BINARY NUMBER IN THE Q-REGISTER AS
29                          ;8 OCTAL DIGITS, TERMINATED BY: SP SP
30                          ; REGISTERS USED: WRK0,WRK1,WRK2
31
32                          TYPAD:                                                  ;TYPEADDRESS:
33 C4155 0631000002CC4006           CALL   F,F,TPOCT                               ; TYPE OCTAL(Q)
34 C4156 0624730350401000           IZQ    OR,1642,0040                            ;
35 C4157 0631000000CC4013           JMP    F,F,TPTXT                               ; TYPETEXT(:<SP><SP>)
```

```
10095 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;TYPE BINARY NUMBER
03
04                          ;THE PROCEDURE TYPE THE BINARY NUMBER IN THE Q-REGISTER AS
05                          ;8 OCTAL DIGITS, TERMINATED BY NL
06                          ; REGISTERS USED: WRK0,WRK1,WRK2
07
08                          TPNUM:                                              ;TYPE NUMBER:
09 C4160 0631000002CC4006           CALL  F,F,TPOCT                            ; TYPE OCTAL(Q)
10 C4161 0631000000CC4007           JMP   F,F,CRLF                             ; TYPE 'CR''LF'
11
12                          ;TYPE 'CR''LF' ON TCP
13
14                          .CRLF:                                              ;
15                          ;CRLF:                                              ;TYPE 'CR''LF'
16 C4162 4624730000CC0320           IZQ   OR,0,13.                             ; Q:= 'CR'
17 C4163 0631000000CC4013           JMP   F,F,TPTXT                            ; GOTO TYPETEXT
18
19                          ;TYPE TEXT
20                          ;
21                          ;THE PROCEDURE TYPES THE THREE CHARACTERS IN THE Q-REGISTER
22                          ; REGISTER USED: WRK0,WRK1
23
24                          .PTXT:                                              ;
25                          ;TPTXT:                                             ;TYPETEXT:
26 C4164 0607340032CC0000           ZBB   AND,WRK0                             ; WRK0:=0
27 C4165 0636330032CC0060           DLZBB OR,WRK0,Z,NL,F,F                     ; WRK0,Q:= WRK0,Q SHIFT 8
28 C4166 0636330032CC0060           DLZBB OR,WRK0,Z,NL,F,F                     ;
29 C4167 0636330032CC0060           DLZBB OR,WRK0,Z,NL,F,F                     ;
30 C4170 0636330032CC0060           DLZBB OR,WRK0,Z,NL,F,F                     ;
31 C4171 0636330032CC0060           DLZBB OR,WRK0,Z,NL,F,F                     ;
32 C4172 0636330032CC0060           DLZBB OR,WRK0,Z,NL,F,F                     ;
33 C4173 0636330032CC0060           DLZBB OR,WRK0,Z,NL,F,F                     ;
34 C4174 0636330032CC0060           DLZBB OR,WRK0,Z,NL,F,F                     ;
35 C4175 4607230033CC0000           ZQB   OR,WRK1                              ; WRK1:=Q
36 C4176 0631000002CC4104           CALL  F,F,TYPE                             ; TYPE(WRK0)
37 C4177 4604431540CC0000           ZAQ   OR,WRK1                              ; Q:= WRK1
38 C4200 6031000000424013           JMP   T,NZ,TPTXT            RTN            ; IF Q<>0 THEN GOTO TYPETEXT
39                                                                             ; ELSE RETURN
```

```
10096 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;PROCEDURE SKIP SPACES UNTIL NL
03                          ;
04                          ;THE PROCEDURE READS A LINE OF SPACES UNTIL 'NL'
05                          ; REGISTER USED: WRK0,Q
06
07                          SKPNL:                                          ;SKIPTONL:
08 04201 4631000002C04010           CALL  F,F,GCHAR                       ; Q:=GETCHAR
09 04202 4665622000CC0240           IQ    SUB0,0,12                       ;
10 04203 2031000000424204           JMP   T,NZ,.+3              RTN       ; IF Q= 'LF' THEN RETURN
11 04204 0665622000CC1000           IQ    SUB0,0,40                       ;
12 04205 4631000000424005           JMP   T,NZ,TCPER                      ; IF Q<>'SP' THEN GOTO TCPERROR
13 04206 4631000000CC4201           JMP   F,F,SKPNL                       ; GOTO SKIPTONL
14
15                          .CPER:                                          ;
16                          ;TCPER:                                         ;TCPERROR:
17 04207 4624700374CC0320           IZQ   OR,"?*16.,13.                   ;
18 04210 4631000002C04013           CALL  F,F,TPTXT                       ; TYPETEXT(?<CR>)
19 04211 4631000000CC0020           JMP   F,F,IDLE                        ; GOTO IDLE
20
21                          ;TYPE <SP><SP>
22
23                          SPSP:                                           ;SPSP:
24 04212 0624730200400000           IZQ   OR,1002,0                       ; Q:=<SP><SP>
25 04213 0631000000CC4013           JMP   F,F,TPTXT                       ; GOTO TYPETEXT
```

```
10097 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;GET REGISTER PROCEDURE
03
04                          ;THE PROCEDURE LOADS WRK0 WITH THE CONTENTS OF THE REGISTER ADDRESSED
05                          ;BY WRK3, FOR NON EXISTING REGISTERS WRK0:=-1
06                          ;REGISTERS USED: Q
07
08                          GETRG:                                                  ;GETGEGISTER:
09 04214 4604431640CC0000           ZAQ    OR,WRK3                                 ; Q:=WRK3
10 04215 0665622000001440           IQ     SUB0,0,62                               ; IF Q=8'62 THEN GOTO GTRTC
11 04216 4631000000024227           JMP    F,NZ,GTRTC                              ;
12 04217 4665622000001400           IQ     SUB0,0,60                               ;
13 04220 0631000000024232           JMP    F,NZ,GDSWT                              ; IF Q=8'60 THEN GOTO GET DATA SWITCH
14 04221 0665622000001420           IQ     SUB0,0,61                               ;
15 04222 4631000000024233           JMP    F,NZ,GRSWT                              ; IF Q=8'61 THEN GOTO GET REG SWITCH
16 04223 4665610000000640           IQ     SUN,0,32                                ;
17 04224 0631000000014234           JMP    F,NEG,DEFLT                             ; IF Q>8'32 THEN GOTO DEFLT
18 04225 0624730000105040           IZQ    OR,0,GRTAB/3                            ; Q:= ADDRESS(GRTAB);
19 04226 4631001641CC0000           JXAQ   ADD,WRK3,F,F                            ; GOTO (Q+WRK3)
20
21
22                          GTRTC:                                                  ;GETRTC:
23 04227 4605431500CC0017           ZAR    OR,WRK0,RTCR                            ;ADDR=62, RTCREAD:=DUMMY
24 04230 4705230000CC0000           ZQ     OR                      CONT 0 2        ;         DELAY MIN 200 NSEC
25 04231 6017730032CC0200           XZB    OR,RTC,WRK0             RTN             ;         WRK0:=RTC
26 04232 6003730032CC1600 GDSWT:    SZB    OR,DATSW,WRK0           RTN             ;ADDR=60, WRK0:= DATSW
27 04233 2003730032CC1700 GRSWT:    SZB    OR,REGSW,WRK0           RTN             ;ADDR=61, WRK0:= REGSW
28
29 04234 6007121532CC0000 DEFLT:    ABB    SUB,WRK0,WRK0           RTN             ;UNKNOWN ADDR, WRK0:= -1
30
31                          VERSION:                                                ;
32 04235 0624730000002020           IZQ    OR,0,65.                                ; WRK0:= IF CAM AVAILABLE THEN 65.
33 04236 0605541400CC5000           RA     AND,SWTCH,.2                            ;
34 04237 4631000000024241           JMP    F,NZ,VERS1                              ;         ELSE 60.
35 04240 0624730000001700           IZQ    OR,0,60.                                ;
36 04241 6007230032CC0000 VERS1:    ZQB    OR,WRK0                 RTN             ;
```

```
10098 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                              GRTAB:                                                  ;GRTAB:
03 C4242 2007431032CC0000              ZAB   OR,W0,WRK0              RTN               ;ADDR= 0, WRK0:=W0
04 C4243 6007431072CC0000              ZAB   OR,W1,WRK0              RTN               ;      1, WRK0:=W1
05 C4244 6007431132CC0000              ZAB   OR,W2,WRK0              RTN               ;      2, WRK0:=W2
06 C4245 2007431172CC0000              ZAB   OR,W3,WRK0              RTN               ;      3, WRK0:=W3
07 C4246 6007431232CC0000              ZAB   OR,STAT,WRK0            RTN               ;      4, WRK0:=STAT
08 C4247 2007730032CC1000              RZB   OR,ICS,WRK0             RTN               ;      5, WRK0:=ICS
09 C4250 2007431272CC0000              ZAB   OR,CAUSE,WRK0           RTN               ;      6, WRK0:=CAUSE
10 C4251 2007431332CC0000              ZAB   OR,SB,WRK0              RTN               ;      7, WRK0:=SB
11 C4252 2003730032CC0000              SZB   OR,CPAC,WRK0            RTN               ;     10, WRK0:=CPAC
12 C4253 6003730032CC0100              SZB   OR,BASEC,WRK0           RTN               ;     11, WRK0:=BASEC
13 C4254 6003730032CC0200              SZB   OR,LLIMC,WRK0           RTN               ;     12, WRK0:=LLIMC
14 C4255 2003730032CC0300              SZB   OR,ULIMC,WRK0           RTN               ;     13, WRK0:=ULIMC
15 C4256 6003730032CC0400              SZB   OR,ILIMC,WRK0           RTN               ;     14, WRK0:=ILIMC
16 C4257 6007431372CC0000              ZAB   OR,INF,WRK0             RTN               ;     15, WRK0:=INF
17 C4260 6007121532CC0000              ABB   SUB,WRK0,WRK0           RTN               ;     16, WRK0:=-1, NOT USED
18 C4261 2003730032CC0500              SZB   OR,SIZE,WRK0            RTN               ;     17, WRK0:=SIZE
19 C4262 2003730032CC0600              SZB   OR,MONT,WRK0            RTN               ;     20, WRK0:=MONTOP
20 C4263 0631000000C04235              JMP   F,F,VERSION                               ;     21, WRK0:=VERSION
21 C4264 6003730032CC1600              SZB   OR,DATSW,WRK0           RTN               ;     22, WRK0:=DATASW C. DATA SWITCH
22 C4265 2003730032CC1700              SZB   OR,REGSW,WRK0           RTN               ;     23, WRK0:=REGSW C. REG SWITCH
23 C4266 6003730032CC0700              SZB   OR,CUR,WRK0             RTN               ;     24, WRK0:=CURRENT PROCES
24 C4267 6003730032CC1000              SZB   OR,PUIX,WRK0            RTN               ;     25, WRK0:=PU INDEX
25 C4270 2003730032CC1100              SZB   OR,PUTAB,WRK0           RTN               ;     26, WRK0:=PUTAB
26 C4271 2003730032CC1200              SZB   OR,EXOFS,WRK0           RTN               ;     27, WRK0:=EXOFS;
27 C4272 6003730032CC1300              SZB   OR,DMOFS,WRK0           RTN               ;     30, WRK0:=DUMP OFFSET
28 C4273 2003730032CC1400              SZB   OR,SWRK0,WRK0           RTN               ;     31, WRK0:=SWRK0
29 C4274 6003730032CC1500              SZB   OR,SWRK1,WRK0           RTN               ;     32, WRK0:=SWRK1
30
```

```
10099 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;PUT REGISTER PROCEDURE
03
04                          ;THE PROCEDURE LOADS THE REGISTER ADDRESSED BY WRK3 WITH THE CONTENTS
05                          ;OF THE WRK0-REGISTER
06
07                          PUTRG:                                                ;PUTREGISTER:
08 04275 4604431640000000           ZAQ    OR,WRK3                               ; Q:=WRK3
09 04276 0665622000001360           IQ     SUB0,0,57                             ; IF Q= 8'57 THEN GOTO PUT57
10 04277 4631000000024307           JMP    F,NZ,PUT57                            ;
11 04300 4665622000001400           IQ     SUB0,0,60                             ;
12 04301 4631000000024310           JMP    F,NZ,PUT60                            ; IF Q= 8'60 THEN GOTO PUT60
13 04302 4665610000000640           IQ     SUN,0,32                              ; IF Q> 8'32 THEN RETURN
14 04303 6031000000414304           JMP    T,NEG,.+3              RTN            ;
15 04304 0637330035000060           SLZBB  OR,WRK3,Z,NL,F,F                      ; WRK3:= WRK3 SHIFT 1
16 04305 4624730000106240           IZQ    OR,0,PRTAB/3                          ; Q:= ADDRESS(PRTAB);
17 04306 4631001641000000           JXAQ   ADD,WRK3,F,F                          ; GOTO (Q+WRK3)
18                          PUT57:                                                ;CLEAR INTERRUPT:
19 04307 6005431500000020           ZAR    OR,WRK0,INTRG          RTN            ;ADDR=57, INTRG:=WRK0
20                          PUT60:                                                ;DISPLAY NUMBER
21 04310 0604431500000000           ZAQ    OR,WRK0                               ;
22 04311 0631000000004160           JMP    F,F,TPNUM                             ; TYPE WRK0 ON TCP
23
24                          PRTAB:                                                ;PRTAB:
25 04312 2007431520000000           ZAB    OR,WRK0,W0             RTN            ;ADDR= 0, W0:=WRK0
26 04313 0605230000000000           ZQ     OR                                    ;
27 04314 6007431521000000           ZAB    OR,WRK0,W1             RTN            ;      1, W1:=WRK0
```

```
!01CC CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 C4315 0605230000CC0000        ZQ    OR                                ;
03 C4316 6007431522CC0000        ZAB   OR,WRKO,W2              RTN      ;      2, W2:=WRKO
04 C4317 0605230000CC0000        ZQ    OR                                ;
05 C4320 2007431523CC0000        ZAB   OR,WRKO,W3              RTN      ;      3, W3:=WRKO
06 C4321 0605230000CC0000        ZQ    OR                                ;
07 C4322 6007431524CC0000        ZAB   OR,WRKO,STAT            RTN      ;      4, STAT:=WRKO
08 C4323 0605230000CC0000        ZQ    OR                                ;
09 C4324 2005431500CC0003        ZAR   OR,WRKO,ICD             RTN      ;      5, ICD:=WRKO
10 C4325 0605230000CC0000        ZQ    OR                                ;
11 C4326 2007431525CC0000        ZAB   OR,WRKO,CAUSE           RTN      ;      6, CAUSE:=WRKO
12 C4327 0605230000CC0000        ZQ    OR                                ;
13 C4330 2007431526CC0000        ZAB   OR,WRKO,SB              RTN      ;      7, SB:=WRKO
14 C4331 0605230000CC0000        ZQ    OR                                ;
15 C4332 4641421500CC0002        ZAS   SUB,WRKO,CPAC                     ;     10, CPAC:=WRKO
16 C4333 6005431500CC0016        ZAR   OR,WRKO,CPA             RTN      ;         CPA:=WRKO
17 C4334 0641421500CC0102        ZAS   SUB,WRKO,BASEC                    ;     11, BASEC:=WRKO
18 C4335 6005431500CC0013        ZAR   OR,WRKO,BASE            RTN      ;         BASE:=WRKO
19 C4336 0641421500CC0202        ZAS   SUB,WRKO,LLIMC                    ;     12, LLIMC:=WRKO
20 C4337 2041513500CC0114        SAR   SUNO,BASEC,WRKO,LLIM    RTN      ;         LLIM:=WRKO - BASEC
21 C4340 4641421500CC0302        ZAS   SUB,WRKO,ULIMC                    ;     13, ULIMC:=WRKO
22 C4341 6041513500CC0115        SAR   SUNO,BASEC,WRKO,ULIM    RTN      ;         ULIM:=WRKO - BASEC
23 C4342 6041421500CC0402        ZAS   SUB,WRKO,ILIMC          RTN      ;     14, ILIMC:=WRKO
24 C4343 0605230000CC0000        ZQ    OR                                ;
25 C4344 6007431527CC0000        ZAB   OR,WRKO,INF             RTN      ;     15, INF:=WRKO
26 C4345 0605230000CC0000        ZQ    OR                                ;
27 C4346 6005230000CC0000        ZQ    OR                      RTN       ;     16, NOT USED
28 C4347 0605230000CC0000        ZQ    OR                                ;
29 C4350 2041421500CC0502        ZAS   SUB,WRKO,SIZE           RTN      ;     17, SIZE:=WRKO
30 C4351 0605230000CC0000        ZQ    OR                                ;
31 C4352 2041421500CC0602        ZAS   SUB,WRKO,MONT           RTN      ;     20, MONT:=WRKO
32 C4353 0605230000CC0000        ZQ    OR                                ;
33 C4354 6005230000CC0000        ZQ    OR                      RTN       ;     21, DUMMY
34 C4355 0605230000CC0000        ZQ    OR                                ;
35 C4356 6041421500CC1602        ZAS   SUB,WRKO,DATSW          RTN      ;     22, DATSW:= WRKO C. DATA SWTCH
36 C4357 0605230000CC0000        ZQ    OR                                ;
```

```
!0101 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 C4360 20414215C0CC1702        ZAS   SUB,WRK0,REGSW          RTN          ;      23, REGSW:= WRK0 C. REG SWTCH
03 C4361 0605230000CC0000        ZQ    OR                                     ;
04 C4362 60414215C0CC0702        ZAS   SUB,WRK0,CUR            RTN          ;      24, CUR:= WRK0
05 C4363 0605230000CC0000        ZQ    OR                                     ;
06 C4364 60414215C0CC1002        ZAS   SUB,WRK0,PUIX           RTN          ;      25, PUIX:= WRK0
07 C4365 0605230000CC0000        ZQ    OR                                     ;
08 C4366 20414215C0CC1102        ZAS   SUB,WRK0,PUTAB          RTN          ;      26, PUTAB:= WRK0
09 C4367 0605230000CC0000        ZQ    OR                                     ;
10 C4370 20414215C0CC1202        ZAS   SUB,WRK0,EXOFS          RTN          ;      27, EXOFS:= WRK0
11 C4371 0605230000CC0000        ZQ    OR                                     ;
12 C4372 60414215C0CC1302        ZAS   SUB,WRK0,DMOFS          RTN          ;      30, DUMP OFS:= WRK0
13 C4373 0605230000CC0000        ZQ    OR                                     ;
14 C4374 20414215C0CC1402        ZAS   SUB,WRK0,SWRK0          RTN          ;      31, SWRK0:= WRK0
15 C4375 0605230000CC0000        ZQ    OR                                     ;
16 C4376 60414215C0CC1502        ZAS   SUB,WRK0,SWRK1          RTN          ;      32, SWRK1:= WRK0
17 C4377 0605230000CC0000        ZQ    OR                                     ;
18
19                               ;EXAMINE REGISTER
20
21                               ;DISPLAYS THE CONTENTS OF THE ADDRESSED REGISTER AS 8
22                               ;OCTAL DIGITS ON THE TCP DISPLAY
23
24                               XR:                                          ;XR:
25 C4400 0631000002C04014              CALL  F,F,GTNUM                       ; WRK2:=GETOCTAL
26 C4401 4613551635CC3500              CAB   CAND,.1B0,WRK2,WRK3             ; WRK3:= 0 EXT 1 CON WRK2(1:23)
27 C4402 4631000002CC4212              CALL  F,F,SPSP                        ; TYPE(<SP><SP>)
28 C4403 4631000002CC4214              CALL  F,F,GETRG                       ; WRK0:=GETREGISTER(WRK3)
29 C4404 0604431500CC0000              ZAQ   OR,WRK0                         ; Q:= WRK0
30 C4405 4631000002CC4160              CALL  F,F,TPNUM                       ; TYPE NUMBER(Q)
31 C4406 4631000000000020              JMP   F,F,IDLE                        ; GOTO IDLE
```

```
1C1C2 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;LOAD REGISTER
03
04                          ;THE ADDRESSED REGISTER IS LOADED WITH THE OCTAL NUMBER
05                          ;TYPED ON THE TCP
06
07                          LR:                                                 ;LR:
08 C4407 0631000002CC4014           CALL  F,F,GTNUM                             ; WRK2:=GETOCTAL
09 C4410 4631000002CC4212           CALL  F,F,SPSP                              ; TYPE(<SP><SP>)
10 C4411 0607431635CC0000           ZAB   OR,WRK2,WRK3                          ; WRK3:=WRK2
11 C4412 0631000002CC4014           CALL  F,F,GTNUM                             ; WRK2:=GETOCTAL
12 C4413 4607431632CC0000           ZAB   OR,WRK2,WRK0                          ; WRK0:=WRK2
13 C4414 0631000002CC4275           CALL  F,F,PUTRG                             ; PUTREGISTER(Q,WRK3)
14 C4415 4631000000CC0020           JMP   F,F,IDLE                              ; GOTO IDLE
15
16                          ;RUN
17
18                          ;STARTS MICROINSTRUCTION EXECUTION FROM ADDRESSED CONTROL
19                          ;STORE LOCATION
20
21                          R:                                                  ;RUN:
22 C4416 0631000002CC4014           CALL  F,F,GTNUM                             ; WRK2:=GETOCTAL
23 C4417 0631431603CC0000           CXZA  OR,WRK2,F,F                           ; CALL (WRK2)
24 C4420 4631000000CC0020           JMP   F,F,IDLE                              ; GOTO IDLE
```

```
101C3 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;LOAD MEMORY
03
04                          ;LOADS THE ADDRESSED MEMORY LOCATION WITH THE OCTAL NUMBER
05                          ;TYPED ON THE TCP
06                          ;THE ADDRESS IS SAVED IN WRK5
07
08                          LM:                                                 ;LM:
09 C4421 0631000002CC4014           CALL  F,F,GTNUM                           ; WRK2:=GETOCTAL
10 C4422 4607431637CC0000           ZAB   OR,WRK2,WRK5                        ; WRK5:=WRK2
11 C4423 4631000002CC4212           CALL  F,F,SPSP                            ; TYPE(<SP><SP>)
12 C4424 4631000000CC4431           JMP   F,F,LDMEM                           ; GOTO LOAD MEM
13
14                          ;LOAD NEXT MEMORY
15
16                          ;LOADS THE MEMORY LOCATION ADDRESSED BY WRK5+2 WITH THE OCTAL
17                          ;NUMBER TYPED ON THE TCP
18                          ;THE ADDRESS IS SAVED IN WRK5
19
20                          LN:                                                 ;LN:
21 C4425 0631000002CC4201           CALL  F,F,SKPNL                           ; SKIP TO NL
22 C4426 4653503777CC3700           CAB   ADDO,.1,WRK5,WRK5                   ; WRK5:= WRK5+2
23 C4427 0604431740CC0000           ZAQ   OR,WRK5                             ; Q:=WRK5
24 C4430 4631000002CC4155           CALL  F,F,TYPAD                           ; TYPE ADDRESS(Q)
25                          LDMEM:                                              ;LOAD MEM:
26 C4431 0631000002CC4014           CALL  F,F,GTNUM                           ; WRK2:=GETOCTAL
27 C4432 4605431600CC0012           ZAR   OR,WRK2,DATO                        ; DATAOUT:=WRK2
28 C4433 0605431740230011           ZAR   OR,WRK5,LAD           CONT WRT      ; LAD:=WRK5, WRT
29 C4434 0654730000CC0100           XZQ   OR,CIOST              CONT U 1      ; Q:=CPUIOSTATUS
30 C4435 4624640000CC0160           IQQ   AND,0,7                             ; IF Q(21:23)<>0 THEN
31 C4436 4631000000424005           JMP   T,NZ,TCPER                          ; GOTO TCP ERROR
32 C4437 4631000000CC0020           JMP   F,F,IDLE                            ; GOTO IDLE
```

```
10104 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;EXAMINE MEMORY
03
04                          ;DISPLAYS THE CONTENTS OF THE ADDRESSED MEMORY LOCATION
05                          ;THE ADDRESS IS SAVED IN WRK5
06
07                          XMM:                                                ;XMM:
08 C4440 0631000002CC4014           CALL  F,F,GTNUM                             ; WRK2:=GETOCTAL
09 C4441 4607431637C00000           ZAB   OR,WRK2,WRK5                          ; WRK5:=WRK2
10 C4442 4631000002CC4455           CALL  F,F,XMEM                              ; EXAMINE(WRK5)
11 C4443 4631000000CC0020           JMP   F,F,IDLE                              ; GOTO IDLE
12
13                          ;EXAMINE NEXT
14
15                          ;DISPLAYS THE CONTENTS OF THE NUMBER OF CONSECUTIVE MEMORY LOCATIONS
16                          ;TYPED ON THE TCP, STARTING AT THE LOCATION ADDRESSED BY WRK5+2
17                          ;THE LAST ADDRESS IS SAVED IN WRK5
18
19                          XN:                                                 ;XN:
20 C4444 0631000002CC4014           CALL  F,F,GTNUM                             ; WRK2:=GETOCTAL
21 C4445 0607431636C00000           ZAB   OR,WRK2,WRK4                          ; WRK4:=WRK2
22 C4446 0631000000C20020           JMP   F,NZ,IDLE                             ; IF WRK4=0 THEN GOTO IDLE
23 C4447 0607101437C00000 XNEXT:    ABB   ADD,.2,WRK5                           ; WRK5:=WRK5+2
24 C4450 4631000002CC4455           CALL  F,F,XMEM                              ; EXAMINE(WRK5)
25 C4451 4607310036C00000           ZBB   SUN,WRK4                              ; WRK4:=WRK4-1
26 C4452 0631000000C20020           JMP   F,NZ,IDLE                             ; IF WRK4=0 THEN GOTO IDLE
27 C4453 4631000000600023           JMP   T,CPUIN,INTIN                         ; IF CPUINTERRUPT THEN GOTO INTIN
28 C4454 0631000000CC4447           JMP   F,F,XNEXT                             ; GOTO XNEXT
```

```
10105 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;DISPLAY MEMORY ADDRESS PROCEDURE
03
04                          ;DISPLAYS THE ADDRESS AND THE CONTENTS OF THE MEMORY LOCATION
05                          ;ADDRESSED BY WRK5
06                          XMEM:                                                ;DISPLAY:
07 C4455 0604431740210011           ZAQR  OR,WRK5,LAD          CONT READ       ; LAD:=Q:=WRK5, READ
08 C4456 4631000002C04155           CALL  F,F,TYPAD                            ; TYPE ADDRESS(Q)
09 C4457 4654730000CC4100           XZQ   OR,IOSTA             CONT 0 1        ; Q:=IOSTATUS
10 C4460 4624640000C00160           IQQ   AND,0,7                              ; IF Q(21:23)<>0 THEN
11 C4461 4631000000424005           JMP   T,NZ,TCPER                           ; GOTO TCP ERROR
12 C4462 4614730000004000           XZQ   OR,DATI                              ; Q:=DATAIN
13 C4463 0631000000CC4160           JMP   F,F,TPNUM                            ; TYPE NUMBER(Q)
```

```
1016 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;SINGLE INSTRUCTION
03
04                          ;EXECUTES THE INSTRUCTION ADDRESSED BY IC AND RETURNS TO IDLE LOOP
05                          ;THE CONTENTS OF W0, W1, W2, W3, STAT, IC, CAUSE, AND SB IS DISPLAYED
06
07                          S:                                                  ;S:
08 C4464 0631000002CC4201           CALL  F,F,SKPNL                           ; SKIP TO NL
09 C4465 4613551471CC1600           CAB   CAND,,1B19,CNTR,CNTR                ; CNTR:= CNTR AND ~,BIT19 (SINGLE INST)
10 C4466 4613531471CC2006           CABR  OR,,1B17,CNTR,CNTR,CNTRO            ; CNTR,CNTRO:= CNTR OR 1B17 (RUN);
11 C4467 4604431200CC0000           ZAQ   OR,STAT                             ; IF STAT = 8'240 THEN
12 C4470 4665622000CC5000           IQ    SUB0,0,0240                         ;
13 C4471 4631000000C22001           JMP   F,NZ,DISPA                          ;   GOTO DISPATCHING POINT;
14 C4472 4631000000CC4500           JMP   F,F,C1                              ; ELSE GOTO C1;
15
16                          ;CONTINUE
17
18                          ;STARTS INSTRUCTION EXECUTION IN THE LOCATION ADDRESSED BY IC,
19                          ; OR FROM THE DISPATCHING POINT
20
21                          C:                                                  ;C:
22 C4473 0631000002CC4201           CALL  F,F,SKPNL                           ; SKIP TO NL
23 C4474 4613551471CC2006           CABR  OR,,1B17,CNTR,CNTR,CNTRO            ; CNTR,CNTRO:= CNTR OR 1B17 (RUN)
24 C4475 4604431200CC0000           ZAQ   OR,STAT                             ; IF STAT = 8'240 THEN
25 C4476 4665622000CC5000           IQ    SUB0,0,0240                         ;
26 C4477 4631000000C22001           JMP   F,NZ,DISPA                          ;   GOTO DISPATCHING POINT;
27                          C1:                                                 ;C1:
28 C4500 4605730000241011           RZR   OR,ICS,LAD              CONT READJ    ; LAD:=ICS, READJ
29 C4501 4613551471CC1506           CABR  CAND,,4,CNTR,CNTR,CNTRO               ; CNTR,CNTRO:= CNTR CAND 1B21;
30 C4502 3455730000410100           XZ    OR,CIOST                NEXT TSTLI 1 ; GOTO NEXT INSTRUCTION
```

```
!01C7 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;EXAMINE DYNAMIC REGISTERS
03
04                          ;DISPLAYS THE CONTENTS OF THE DYNAMIC REGISTERS
05
06                          XD:                                                 ;XD:
07 C4503 0631000002C04201           CALL  F,F,SKPNL                           ; SKIP TO NL
08 C4504 4631000002C04515           CALL  F,F,TPDYN                           ; TYPE DYNAMIC REGISTERS
09 C4505 4631000000C00020           JMP   F,F,IDLE                            ; GOTO IDLE
10
11                          ;EXAMINE STATIC REGISTERS
12
13                          ;DISPLAYS THE CONTENTS OF THE STATIC REGISTERS
14
15                          XST:                                                ;XS:
16 C4506 0631000002C04201           CALL  F,F,SKPNL                           ; SKIP TO NL
17 C4507 4631000002C04540           CALL  F,F,TPSTA                           ; TYPE STAT. REGISTERS
18 C4510 4631000000C00020           JMP   F,F,IDLE                            ; GOTO IDLE
19
20
21                          ;SINGLE INSTRUCTION INTERRUPT
22
23                          .INGL:                                              ;
24                          ;SINGL:                                             ;SINGLE INSTR. INT:
25 C4511 4613551471002000           CAB   CAND,.1B17,CNTR,CNTR                ; CNTR:=CNTR CAND BIT17 (RUN:=0)
26 C4512 4613551471001606           CABR  OR,.1B19,CNTR,CNTR,CNTRO            ;CNTR:=CNTRO:=CNTR OR BIT19
27                                                                            ; (=,SINGLE INSTR INT:=1)
28 C4513 4631000002C04515           CALL  F,F,TPDYN                           ; TYPE DYNAMIC REGISTERS
29 C4514 4631000000C00020           JMP   F,F,IDLE                            ; GOTO IDLE
```

```
!01C8 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;TYPE DYNAMIC REGISTERS
03
04                          ;THE PROCEDURE TYPES THE CONTENTS OF THE 8 DYNAMIC REGISTERS
05
06                          TPDYN:                                                  ;TYPE DYN:
07 C4515 4607340035CC0000           ZBB   AND,WRK3                                  ; FOR WRK3:=0 STEP 1 UNTIL 7 DO
08 C4516 4624730000112600 TPDY1:    IZQ   OR,U,DYTXT/3                              ; BEGIN
09 C4517 0631001643CC0000           CXAQ  ADD,WRK3,F,F                              ;   Q:=ADDRESS(DYTXT)
10 C4520 4631000002CC4013           CALL  F,F,TPTXT                                 ;   Q:=REGTXT(Q+WRK3)
11 C4521 4631000002CC4212           CALL  F,F,SPSP                                  ;   TYPE TEXT(Q)
12 C4522 4631000002CC4214           CALL  F,F,GETRG                                 ;   TYPE(<SP><SP>)
13 C4523 0604431500CC0000           ZAQ   OR,WRK0                                   ;   WRK0:=GETREGISTER(WRK3)
14                                                                                  ;   Q:= WRK0
15 C4524 4631000002CC4160           CALL  F,F,TPNUM                                 ;
16 C4525 4607302035CC0000           ZBB   ADD0,WRK3                                 ;   TYPE NUMBER(Q)
17 C4526 4651523640CC3400           CA    SUB0,,7,WRK3                              ; END;
18 C4527 2031000000014516           JMP   F,NEG,TPDY1             RTN               ;
19
20                          DYTXT:                                                  ;REGISTER TEXT
21 C4530 2024730534601640           IZQ   OR,2563,0072            RTN               ; Q:='W0:'
22 C4531 6024730534611640           IZQ   OR,2563,0472            RTN               ; Q:='W1:'
23 C4532 6024730534621640           IZQ   OR,2563,1072            RTN               ; Q:='W2:'
24 C4533 2024730534631640           IZQ   OR,2563,1472            RTN               ; Q:='W3:'
25 C4534 2024730515241640           IZQ   OR,2465,2072            RTN               ; Q:='ST:'
26 C4535 6024730445031640           IZQ   OR,2224,1472            RTN               ; Q:='IC:'
27 C4536 2024730415231640           IZQ   OR,2065,1472            RTN               ; Q:='CS:'
28 C4537 6024730515021640           IZQ   OR,2464,1072            RTN               ; Q:='SB:'
```

```
1010C9 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;TYPE STATIC REGISTERS
03
04                          ;THE PROCEDURE TYPES THE CONTENTS OF THE STATIC REGISTERS
05
06                          TPSTA:                                                  ;
07 C4540 4607340036CC0000           ZBB   AND,WRK4                                  ; FOR WRK4:=0 STEP 1 UNTIL 8 DO
08 C4541 4613730035C03600           CZB   OR,.8,WRK3                                ; BEGIN
09 C4542 0624730000113300 TPST1:    IZQ   OR,0,STTXT/3                              ;   Q:=ADDRESS(STTXT);
10 C4543 0631001703CC0000           CXAQ  ADD,WRK4,F,F                              ;   Q:=REGTXT(WRK4+Q)
11 C4544 4631000002CC4013           CALL  F,F,TPTXT                                 ;   TYPE TEXT(Q)
12 C4545 4631000002CC4212           CALL  F,F,SPSP                                  ;   TYPE(<SP><SP>)
13 C4546 4631000002CC4214           CALL  F,F,GETRG                                 ;   WRK0:=GETREG(WRK4+8)
14 C4547 0604431500CC0000           ZAQ   OR,WRK0                                   ;   Q:= WRK0
15 C4550 4631000002CC4160           CALL  F,F,TPNUM                                 ;   TYPE NUMBER(Q)
16 C4551 4607302035CC0000 TPST2:    ZBB   ADD0,WRK3                                 ;
17 C4552 4607302036CC0000           ZBB   ADD0,WRK4                                 ;
18 C4553 4631000000CC4542           JMP   F,F,TPST1                                 ; END
19
20                          STTXT:                                                  ;REGISTER TEXT
21 C4554 2024700415201640           IZQ   OR,"C*16.+("P/16.),"P*256.+":  RTN        ; 10: Q:='CP:'
22 C4555 6024700411011640           IZQ   OR,"B*16.+("A/16.),"A*256.+":  RTN        ; 11: Q:='BA:'
23 C4556 6024700461141640           IZQ   OR,"L*16.+("L/16.),"L*256.+":  RTN        ; 12: Q:='LL:'
24 C4557 2024700525141640           IZQ   OR,"U*16.+("L/16.),"L*256.+":  RTN        ; 13: Q:='UL:'
25 C4560 6024700445141640           IZQ   OR,"I*16.+("L/16.),"L*256.+":  RTN        ; 14: Q:='IL:'
26 C4561 2024700445161640           IZQ   OR,"I*16.+("N/16.),"N*256.+":  RTN        ; 15: Q:='IN:'
27 C4562 0631000000CC4575           JMP   F,F,TPSKP                                 ; 16: DUMMY
28 C4563 2024700515111640           IZQ   OR,"S*16.+("I/16.),"I*256.+":  RTN        ; 17: Q:='SI:'
29 C4564 2024700465171640           IZQ   OR,"M*16.+("O/16.),"O*256.+":  RTN        ; 20: Q:='MO:'
30 C4565 0631000000CC4575           JMP   F,F,TPSKP                                 ; 21: DUMMY
31 C4566 0631000000CC4575           JMP   F,F,TPSKP                                 ; 22: DUMMY
32 C4567 0631000000CC4575           JMP   F,F,TPSKP                                 ; 23: DUMMY
33 C4570 2024700415251640           IZQ   OR,"C*16.+("U/16.),"U*256.+":  RTN        ; 24: Q:='CU:'
34 C4571 2024700501301640           IZQ   OR,"P*16.+("X/16.),"X*256.+":  RTN        ; 25: Q:='PX:'
35 C4572 2024700501241640           IZQ   OR,"P*16.+("T/16.),"T*256.+":  RTN        ; 26: Q:='PT:'
36 C4573 0005230000CC0000           ZQ    OR                             POP    ;
37 C4574 6005230000CC0000           ZQ    OR                             RTN    ;
38
39 C4575 0005230000CC0000 TPSKP:    ZQ    OR                             POP    ;
40 C4576 0631000000CC4551           JMP   F,F,TPST2                              ;
```

```
!0110 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;EXAMINE WORK REGISTERS
03
04                          ;DISPLAYS THE CONTENTS OF THE WORKING REGISTERS
05                          XW:                                                 ;XW:
06 04577 0631000002004201           CALL   F,F,SKPNL                            ; SKIP TO NL
07 04600 4641421540001502           ZAS    SUB,WRK1,SWRK1                       ; SAVE WRK1 IN SWRK1
08 04601 0641421600001602           ZAS    SUB,WRK2,DATSW                       ; SAVE WRK2 IN DATSW
09 04602 0641421640001702           ZAS    SUB,WRK3,REGSW                       ; SAVE WRK3 IN REGSW
10 04603 0207340035000000           ZBB    AND,WRK3                PUSH         ; FOR WRK3:= 0,WRK3+1 WHILE TRUE DO
11 04604 0624730000114200           IZQ    OR,0,WRKREG/3                        ; BEGIN
12 04605 0631001643000000           CXAQ   ADD,WRK3,F,F                         ;   Q:= GETWORK(Q,WRK3)
13 04606 4631000002004160           CALL   F,F,TPNUM                            ;   TYPENUMBER(Q)
14 04607 2637302035404060           NSZBB  ADDO,WRK3,NL,T,F        LRTN         ; END
15
16                          WRKREG:                                             ;WRKREG:
17 04610 6000730000001400           SZQ    OR,SWRK0                RTN          ; Q:= SWRK0
18 04611 2000730000001500           SZQ    OR,SWRK1                RTN          ; Q:= SWRK1
19 04612 2000730000001600           SZQ    OR,DATSW                RTN          ; Q:= DATSW
20 04613 6000730000001700           SZQ    OR,REGSW                RTN          ; Q:= REGSW
21 04614 2004431700000000           ZAQ    OR,WRK4                 RTN          ; Q:= WRK4
22 04615 6004431740000000           ZAQ    OR,WRK5                 RTN          ; Q:= WRK5
23 04616 4631000000000020           JMP    F,F,IDLE                             ;
```

```
10111 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;POWER UP
03                          ;********
04
05                          .OWUP:                                                  ;
06                          ;POWUP:                                                 ;POWUP:
07 04617 4624730000CC0760           IZQ    OR,0,37                                 ; CNTR:=CNTRO:= 8'37
08 04620 0607230031CC0006           ZQBR   OR,CNTR,CNTRO                           ;
09 04621 0613551471C03606           CABR   CAND,,1B20,CNTR,CNTR,CNTRO              ; RESET REMOTE AUTOLOAD
10 04622 0613531471CC3606           CABR   OR,,1B20,CNTR,CNTR,CNTRO                ;
11 04623 4631000002CC5044           CALL   F,F,INITR                               ; INIT REGISTERS
12 04624 4624730000CC2000           IZQ    OR,0,100                                ; IF SWITCH(17)<>0 THEN
13 04625 0605640000CC5000           RQ     AND,SWTCH                               ; BEGIN  C. POWER UP TEST
14 04626 0631000000C25002           JMP    F,NZ,POWU1                              ;   TEST THE THREE ERROR LAMPS
15 04627 0624730000CC0600           IZQ    OR,0,30                                 ;
16 04630 0605230000CC0006           ZQR    OR,CNTRO                                ;
17 04631 0605230000C00000 CLLOP:    ZQ     OR                      CONT 0 0        ;   REPEAT
18 04632 4645230000CC0000           ZQ     OR                      CONT 0 1        ;     CPUCLOCK TEST 0,1,2,3
19 04633 4705230000CC0000           ZQ     OR                      CONT 0 2        ;   UNTIL OCP-AUTOLOAD
20 04634 0745230000CC0000           ZQ     OR                      CONT 0 3        ;
21 04635 0631000000654631           JMP    T,NOCPA,CLLOP                           ;
22 04636 0631000002CC4644           CALL   F,F,LAMP                                ;   TURN ERROR LAMPS OFF
23 04637 4631000000254637           JMP    F,NOCPA,.                               ;   WAIT FOR AUTOLOAD PB TO BE RELEASED
24 04640 0624731252525240 IMLOP:    IZQ    OR,5252,5252                            ;
25 04641 0624670525252520           IQQ    EXNOR,2525,2525                         ;   C. TEST OF IMOP
26                                                                                  ;   Q:= 2'1010...010
27                                                                                  ;   Q:= Q -,EXOR 2'0101....101
28                                                                                  ;   IF Q<>0 THEN
29 04642 0631000000424646           JMP    T,NZ,IMOER                              ;     GOTO IMOP-ERROR
30 04643 0631000000CC4662           JMP    F,F,ARLOT                               ;   GOTO ARITHMETIC-LOGICTEST
31 04644 4624730000CC0760 LAMP:     IZQ    OR,0,37                                 ; TURN ERROR LAMPS OFF
32 04645 6005230000C00006           ZQR    OR,CNTRO                RTN             ;   RETURN
```

```
10112 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                              IMOER:                                          ;IMOPERROR:
03 C4646 0624730000CC0740              IZQ     OR,0,36                         ;   TURN ERRORLAMP 3 ON
04 C4647 0605230000CC0006              ZQR     OR,CNTRO                        ;
05 C4650 0631000000255002              JMP     F,NOCPA,POWU1                   ;   IF OCPAUTOLOAD THEN GOTO POWU1
06 C4651 0631000000CC4640              JMP     F,F,IMLOP                       ;   GOTO IMLOP
07                              ARLOER:                                         ;ARITHMETIC-LOGIC ERROR:
08 C4652 0624730000CC0720              IZQ     OR,0,35                         ;   TURN ERRORLAMP 2 ON
09 C4653 0605230000CC0006              ZQR     OR,CNTRO                        ;
10 C4654 0631000000255002              JMP     F,NOCPA,POWU1                   ;   IF OCPAUTOLOAD THEN GOTO POWU1
11 C4655 0631000000CC4662              JMP     F,F,ARLOT                       ;
12                              ALUER:                                          ;ALUERROR:
13 C4656 4624730000CC0700              IZQ     OR,0,34                         ;   TURN ERRORLAMPS 2,3 ON
14 C4657 0605230000CC0006              ZQR     OR,CNTRO                        ;
15 C4660 0631000000255002              JMP     F,NOCPA,POWU1                   ;   IF OCPAUTOLOAD THEN GOTO POWU1
16 C4661 0631000000CC4727              JMP     F,F,ALUTS                       ;   GOTO ALUTS
17
18                              ARLOT:                                          ;ARITHMETIC-LOGIC TEST:
19 C4662 0664703777777760              IZQ     ADDO,7777,7777                  ;   TEST (-1)+0 +1
20 C4663 0631000000424652              JMP     T,NZ,ARLOE                      ;
21 C4664 0664711777777760              IZQ     SUN,7777,7777                   ;   TEST -(-1)+0 -1
22 C4665 0631000000424652              JMP     T,NZ,ARLOE                      ;
23 C4666 4624730000CC0020              IZQ     OR,0,1                          ;
24 C4667 4664622000CC0020              IQQ     SUBO,0,1                        ;   TEST 1-(1)
25 C4670 0631000000424652              JMP     T,NZ,ARLOE                      ;
26 C4671 0624730000CC0000              IZQ     OR,0,0                          ;   TEST 0 OR 0
27 C4672 0631000000424652              JMP     T,NZ,ARLOE                      ;
28 C4673 0624731777777760              IZQ     OR,7777,7777                    ;   TEST -1 OR 0
29 C4674 4605202000CC0000              ZQ      ADDO                            ;
30 C4675 0631000000424652              JMP     T,NZ,ARLOE                      ;
31 C4676 4624630000CC0000              IQQ     OR,0,0                          ;   TEST 0 OR -1
32 C4677 4605202000CC0000              ZQ      ADDO                            ;
33 C4700 0631000000424652              JMP     T,NZ,ARLOE                      ;
34 C4701 4624631777777760              IQQ     OR,7777,7777                    ;   TEST -1 OR -1
35 C4702 4605202000CC0000              ZQ      ADDO                            ;
36 C4703 0631000000424652              JMP     T,NZ,ARLOE                      ;
```

```
10113 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 C4704 4624740000000000          IZQ   AND,0,0                    ;   TEST 0 AND 0
03 C4705 0631000000424652          JMP   T,NZ,ARLOE                 ;
04 C4706 4624741777777760          IZQ   AND,7777,7777              ;   TEST -1 AND 0
05 C4707 0631000000424652          JMP   T,NZ,ARLOE                 ;
06 C4710 0624731777777760          IZQ   OR,7777,7777               ;
07 C4711 0624640000000000          IQQ   AND,0,0                    ;   TEST 0 AND -1
08 C4712 0631000000424652          JMP   T,NZ,ARLOE                 ;
09 C4713 0624731777777760          IZQ   OR,7777,7777               ;
10 C4714 0624641777777760          IQQ   AND,7777,7777              ;   TEST -1 AND -1
11 C4715 4605202000000000          ZQ    ADDO                       ;
12 C4716 0631000000424652          JMP   T,NZ,ARLOE                 ;
13 C4717 0624750000000000          IZQ   CAND,0,0                   ;   TEST -,0 AND 0
14 C4720 0631000000424652          JMP   T,NZ,ARLOE                 ;
15 C4721 0624751777777760          IZQ   CAND,7777,7777             ;   TEST -, -1 AND 0
16 C4722 0631000000424652          JMP   T,NZ,ARLOE                 ;
17 C4723 0624731777777760          IZQ   OR,7777,7777               ;
18 C4724 4624650000000000          IQQ   CAND,0,0                   ;   TEST -, 0 AND -1
19 C4725 4605202000000000          ZQ    ADDO                       ;
20 C4726 0631000000424652          JMP   T,NZ,ARLOE                 ;
21                         ALUTS:                                   ; ALU TEST:
22 C4727 4624730000000020          IZQ   OR,0,1                     ;   Q:= 1
23                         ALURG:                                   ;  ALU-REGISTER TEST:
24 C4730 0607230032000000          ZQB   OR,WRK0                    ;   WHILE Q<>0 DO
25 C4731 0607023532000000          AQB   SUBO,WRK0,WRK0             ;   BEGIN
26 C4732 4631000000424656          JMP   T,NZ,ALUER                 ;
27 C4733 4607230033000000          ZQB   OR,WRK1                    ;     FOR REG:= WRK0,..,WRK5 DO
28 C4734 0607023573000000          AQB   SUBO,WRK1,WRK1             ;     BEGIN
29 C4735 4631000000424656          JMP   T,NZ,ALUER                 ;
30 C4736 0607230034000000          ZQB   OR,WRK2                    ;       REG:= Q
31 C4737 0607023634000000          AQB   SUBO,WRK2,WRK2             ;       IF Q-REG<>0 THEN
32 C4740 4631000000424656          JMP   T,NZ,ALUER                 ;         ERROR RETURN
33 C4741 4607230035000000          ZQB   OR,WRK3                    ;
34 C4742 0607023675000000          AQB   SUBO,WRK3,WRK3             ;     END
35 C4743 4631000000424656          JMP   T,NZ,ALUER                 ;
36 C4744 4607230036000000          ZQB   OR,WRK4                    ;
37 C4745 0607023736000000          AQB   SUBO,WRK4,WRK4             ;
38 C4746 4631000000424656          JMP   T,NZ,ALUER                 ;
```

```
10114 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 C4747 0607230037CC0000        ZQB   OR,WRK5                          ;
03 C4750 0607023777CC0000        AQB   SUBO,WRK5,WRK5                   ;
04 C4751 4631000000424656        JMP   T,NZ,ALUER                       ;
05 C4752 0636330032CC0060        DLZBB OR,WRKO,Z,NL,F,F                 ;     Q:= Q SHIFT 1
06 C4753 0605230000CC0000        ZQ    OR                               ;
07 C4754 0631000000424730        JMP   T,NZ,ALURG                       ;   END;
08 C4755 0605411500CC0000        ZA    SUN,WRKU                         ;   IF WRKU<>1 THEN
09 C4756 4631000000424656        JMP   T,NZ,ALUER                       ;     GOTO ALUER C. TEST DOUBLE SHIFT
10 C4757 0631000000654640        JMP   T,NOCPA,IMLOP                    ;   IF NOT OCPAUTOL THEN GOTO IMLOP
11 C4760 4631000000254760        JMP   F,NOCPA,.                        ;   WAIT UNTIL OCP-PB IS RELEASED
12                                                                      ;   C. TEST OF TCP
13 C4761 4624700521312400        IZQ   OR,"T*16.+("Y/16.),"Y*256.+"P    ;
14 C4762 4631000002CC4013        CALL  F,F,TPTXT                        ;   TYPE "TYPE OCTAL NUMBER:"
15 C4763 4624700424402360        IZQ   OR,"E*16.+(32./16.),32.*256.+"O  ;   ON THE TCP
16 C4764 4631000002CC4013        CALL  F,F,TPTXT                        ;
17 C4765 0624700415242020        IZQ   OR,"C*16.+("T/16.),"T*256.+"A    ;
18 C4766 4631000002CC4013        CALL  F,F,TPTXT                        ;
19 C4767 0624700460402340        IZQ   OR,"L*16.+(32./16.),32.*256.+"N  ;
20 C4770 4631000002CC4013        CALL  F,F,TPTXT                        ;
21 C4771 0624700525152040        IZQ   OR,"U*16.+("M/16.),"M*256.+"B    ;
22 C4772 4631000002CC4013        CALL  F,F,TPTXT                        ;
23 C4773 0624700425221640        IZQ   OR,"E*16.+("R/16.),"R*256.+":    ;
24 C4774 4631000002CC4013        CALL  F,F,TPTXT                        ;
25 C4775 0624730000CC0660        IZQ   OR,0,33                          ;   TURN LAMPS 1 ON
26 C4776 0605230000CC0006        ZQR   OR,CNTRO                         ;
27 C4777 4631000002CC4014        CALL  F,F,GTNUM                        ;   WRK2:= READ OCTAL NUMBER
28 C5000 0604431600CC0000        ZAQ   OR,WRK2                          ;   WRK2:= Q
29 C5001 4631000002CC4160        CALL  F,F,TPNUM                        ;   TYPE OCTAL NUMBER(Q)
```

```
!0115 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          POWU1:                                          ; END  C. CONTINUE WITH POWER UP
03 C5002 4613730024CC3505           CZBR  OR,.1BU,STAT,CPUST                ; STAT:=CPUSI:= 1BU , MONMUDE
04 C5003 4631000000675003           JMP   T,PLOW,.                          ; WAIT UNTIL NOT POWER LOW
05 C5004 062473U140CC0000           IZQ   OR,600,0                          ;
06 C5005 0631000002CC4012           CALL  F,F,DELAY                         ; DELAY(600 MSEC) FOR MEMORY POWER ON
07 C5006 0631000002CC4016           CALL  F,F,CINTR                         ; CLEAR ALL INTERRUPTS
08 C5007 0631000002CC5347           CALL  F,F,CAMIN                         ; INIT CAM
09 C5010 4631000000C255030          JMP   F,NOCPA,NREST                     ; IF -,OCPAUTOLOAD OR SWITCH(16) THEN
10 C5011 4604730000CC5000           RZQ   OR,SWTCH                          ; BEGIN  C. RESTART
11 C5012 4624640000CCC4000          IQQ   AND,0,200                         ;
12 C5013 4631000000C25030           JMP   F,NZ,NREST                        ;
13 C5014 4614730000CCC0100          XZQ   OR,CIOST                          ;
14 C5015 0625640076CCC0000          IQ    AND,037U,0                        ;   IF DEVICENUMBER=0 THEN
15 C5016 0631000000425030           JMP   T,NZ,NREST                        ;   BEGIN
16 C5017 0611730000201311           CZR   OR,.10,LAD              CONT READP ;     LAD:= 10  , READP
17 C5020 4654730000CC4100           XZQ   OR,IOSTA                CONT U 1  ;
18 C5021 4624640000CCC0160          IQQ   AND,U,7                           ;     IF IOSTATUS(21:23)<>0 THEN
19 C5022 0631000000425237           JMP   T,NZ,SYFL3                        ;       GOTO SYSTEMFAULT3
20 C5023 0615730000CC4003           XZR   OR,DATI,ICD                       ;     ICD:= DATAIN
21 C5024 0615730000244011           XZR   OR,DATI,LAD             CONT READJ ;     LAD:= DATAIN  , READ INSTRUCTION
22 C5025 4613531471CC2U06           CABR  OR,.1B17,CNTR,CNTR,CNTRO          ;     RUN:= 1
23 C5026 4613551471CC1506           CABR  CAND,.4,CNTR,CNTR,CNTRO           ;
24 C5027 7405230000CCC0000          ZQ    OR                      NEXT      ;     GOTO NEXT INSTR
25                                                                          ; END; END;
26                          NREST:                                          ;NOT RESTART:
27                                                                          ;  C. OCP AUTOLOAD OR SEM COND MEMORY
28 C5030 4613531471CC2006           CABR  OR,.1B17,CNTR,CNTR,CNTRO          ; RUN:=1
29 C5031 0631000000255031           JMP   F,NOCPA,.                         ; WAIT UNTIL AUTOLOAD PB IS RELEASED
30 C5032 0624730002CC0200           IZQ   OR,8.,8.                          ;
31 C5033 0605230000CC0021           ZQR   OR,ILIM                           ; ILIM:= 8.; ENABLE START
32 C5034 4631000000CC0020           JMP   F,F,IDLE                          ; GOTO IDLE
```

```
10116 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          .ELAY:                                          ;
03                          ;DELAY:                                         ;DELAY:
04 C5035 0604210000000000           ZQQ   SUN                               ; REPEAT
05 C5036 6031000000425035           JMP   T,NZ,.-3              RTN         ;   Q:= Q-1
06                                                                          ; UNTIL Q=0
07                          ;CINTR:                                         ;CLEAR INTERRUPT:
08                          .INTR:                                          ;
09 C5037 4624730000002000           IZQ   OR,0,100                          ; Q:= 64
10 C5040 0604210000000000 CINT1:    ZQQ   SUN                               ; REPEAT
11 C5041 4605230000000020           ZQR   OR,INTRG                          ;   Q:= Q-1
12 C5042 0651622000003000           CQ    SUB0,.6                           ;   CLEAR INTERRUPT(Q)
13 C5043 2031000000425040           JMP   T,NZ,CINT1            RTN         ; UNTIL Q=6
14
15                          INITR:                                          ;INIT REG:
16 C5044 4641121532000102           ABS   SUB,WRK0,WRK0,BASEC               ; BASEC:= 0
17 C5045 0605240000000013           ZQR   AND,BASE                          ; BASE:= 0
18 C5046 4641121532000402           ABS   SUB,WRK0,WRK0,ILIMC               ; ILIMC:= 0
19 C5047 4605240000000021           ZQR   AND,ILIM                          ; ILIM:= 0;
20 C5050 4641121532001002           ABS   SUB,WRK0,WRK0,PUIX                ; PUIX:= 0;
21 C5051 0601240000000702           ZQS   AND,CUR                           ; CUR:= -1
22 C5052 0613730027003700           CZB   OR,.1,INF                         ; INF:= 1
23 C5053 0607403370000000           ZAB   ADD0,INF,.2                       ; .2:= 2
24 C5054 4624730000000200           IZQ   OR,0,10                           ;
25 C5055 4641220000000202           ZQS   SUB,LLIMC                         ; LLIMC:=LLIM:= 8
26 C5056 0605230000000014           ZQR   OR,LLIM                           ;
27 C5057 4624730777777760           IZQ   OR,3777,7777                      ;
28 C5060 0641220000000302           ZQS   SUB,ULIMC                         ; ULIMC:=ULIM:= 8'37777777
29 C5061 4605230000000015           ZQR   OR,ULIM                           ;
30 C5062 4605240000000024           ZQR   AND,CAMBC                         ; CAMBC:= 0;
31 C5063 0601240000001602           ZQS   AND,DATSW                         ;
32 C5064 2001240000001702           ZQS   AND,REGSW             RTN         ; DATASW,REGSW:= -1
```

```
10117 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          SYSRS:                                          ;SYSTEM RESET:
03 C5065 0613531471CC2106           CABR   OR,.1B16,CNTR,CNTR,CNTRO         ; CPU SYSTEM RESET:= 1
04 C5066 0624730000CC0560           IZQ    OR,0,27                          ;
05 C5067 0631000002CC4012           CALL   F,F,DELAY                        ; DELAY(9 MICROSEC)
06 C5070 0613531471CC2106           CABR   CAND,.1B16,CNTR,CNTR,CNTRO       ; CPU SYSTEM RESET:= 0
07 C5071 4624730000CC7660           IZQ    OR,0,373                         ;
08 C5072 4631000000CC4012           JMP    F,F,DELAY                        ; DELAY(100 MICROSEC)
```

```
10118 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;AUTOLOAD SEQUENCES
03                          ;******************
04
05                          ;OCP AUTOLOAD
06
07                          .CPAL:                                          ;
08                          ;OCPAL:                                         ;OCP AUTOLOAD:
09 C5073 4653720031002006           CZBR  SUB,.64,CNTR,CNTRO                ; CNTR:=CNTRO:= 8'77
10 C5074 4631000000255074           JMP   F,NOCPA,.                         ; WAIT UNTIL AUTOLOAD PB IS RELEASED
11 C5075 4614730000000100           XZQ   OR,CIOST                          ; IF DEVICENO <> 0 THEN
12 C5076 0625640076000000           IQ    AND,0370,0                        ;  GOTO POWUP;
13 C5077 4631000000424000           JMP   T,NZ,POWUP                        ; <* FIRST PU DOES THE AUTOLOAD *>
14 C5100 4631000002CC5065           CALL  F,F,SYSRS                         ; SYSTEM RESET
15 C5101 4624730000000200           IZQ   OR,0,10                           ; IF FRONT SWITCH=1 THEN
16 C5102 0605640000005000           RQ    AND,SWTCH                         ;   INIT CAM
17 C5103 4631000000024015           JMP   F,NZ,RAUTO                        ;   GOTO AUTOTEST
18 C5104 4631000002C05347           CALL  F,F,CAMIN                         ;
19 C5105 4631000000006001           JMP   F,F,ATEST                         ;
20                          .AUTO:                                          ;
21                          ;RAUTO:                                         ;RETURN FROM AUTOLOADTEST
22 C5106 4613531471CC3406           CABR  OR,.7,CNTR,CNTR,CNTRO             ; CLEAR ERROR LAMPS
23 C5107 4624731777777300           IZQ   OR,7777,7754                      ; Q:= 14-4*10 , DEV BASE FOR DISC
24 C5110 0631000000005115           JMP   F,F,AUTOL                         ; GOTO COMMON AUTOLOAD SEQUENCE
25
26                          ;REMOTE AUTOLOAD
27
28                          .EMAL:                                          ;
29                          ;REMAL:                                         ;REMOTE AUTOLOAD:
30 C5111 0613551471CC3606           CABR  CAND,.1820,CNTR,CNTR,CNTRO        ; RESET REMOTE AUTOLOAD
31 C5112 0613531471CC3606           CABR  OR,.1820,CNTR,CNTR,CNTRO          ;
32 C5113 4631000002CC5065           CALL  F,F,SYSRS                         ; SYSTEM RESET
33 C5114 0624731777777700           IZQ   OR,7777,7774                      ; Q:=14 - 2*10 , DEV BASE FOR FPA
```

```
10119 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          AUTOL:                                          ;COMMON AUTOLOAD SEQUENCE:
03 C5115 0607230034CC0000           ZQB   OR,WRK2                          ; WRK2:=Q, DEV BASE
04 C5116 4631000002CC5044           CALL  F,F,INITR                       ; INIT REGISTERS
05 C5117 4631000002CC5347           CALL  F,F,CAMIN                       ; INIT CAM
06 C5120 4613730024CC1705           CZBR  OR,.1B18,STAT,CPUST             ; STAT,CPUST:= 1B18; NO PROCESS ACTIVE
07 C5121 0624730000C20540           IZQ   OR,0,1026                       ;
08 C5122 4605230000CC0016           ZQR   OR,CPA                          ; CPA:=CPAC:= 512.+22.
09 C5123 0641220000CC0002           ZQS   SUB,CPAC                        ;
10 C5124 4605230000CC0015           ZQR   OR,ULIM                         ;
11 C5125 0641220000CC0302           ZQS   SUB,ULIMC                       ; ULIM:=ULIMC:= 521.+22.
12 C5126 0605230000CC0014           ZQR   OR,LLIM                         ;
13 C5127 4641220000CC0202           ZQS   SUB,LLIMC                       ; LLIM:=LLIMC:= 512.+22.
14 C5130 4605013400CC0003           AQR   SUNO,.2,ICD                     ; ICD:= 521.+22.-2
15 C5131 0624730000C20320           IZQ   OR,0,1015                       ;
16 C5132 4607230027CC0000           ZQB   OR,INF                          ; INF:= 512.+22.-9.
17 C5133 0624730777777740           IZQ   OR,3777,7776                    ;
18 C5134 0641220000CC0502           ZQS   SUB,SIZE                        ; SIZE:= 8'37777776
19 C5135 4624730000C77760           IZQ   OR,0,3777                       ;
20 C5136 0641220000CC0602           ZQS   SUB,MONT                        ; MONTOP:= 8'3777
21
22 C5137 4613730037CC3400           CZB   OR,.7,WRK5                      ; WRK5:= 7
23 C5140 4613730033CC3600           CZB   OR,.8,WRK1                      ; WRK1:= 8;
24 C5141 0607340032CC0000           ZBB   AND,WRK0                        ; WRK0:= 0
25 C5142 4251712000CC3004           CZR   SUNO,.6,LCD           PUSH      ; FOR LCD:= -6 STEP 1 UNTIL 0 DO
26 C5143 4624730000124300           IZQ   OR,0,BOOTL/3                    ; BEGIN
27 C5144 4631001503CC0000           CXAQ  ADD,WRK0,F,F                    ;   Q:=ADDR(BOOTL+WRK0)
28 C5145 0605230000CC0012           ZQR   OR,DATO                         ;   DATAOUT:= Q
29 C5146 4647101433230011           ABBR  ADD,.2,WRK1,LAD       CONT WRT  ;   LAD:=WRK1:= WRK1+2  , WRT
30 C5147 0655541740CC0100           XA    AND,CIOST,WRK5        CONT 0 1  ;   IF CPUIOSTATUS(21:23)<>0 THEN
31 C5150 4631000000425241           JMP   T,NZ,SYFL4                      ;     GOTO SYSTEMFAULT4
32 C5151 2637302032114060           NSZBB ADDO,WRK0,NL,F,LCN    LRTN      ;   WRK0:= WRK0+1
33                                                                        ; END
34
```

```
10120 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 C5152 0624730322CC0000          IZQ   OR,1510,0                          ; Q:= 8'15100000 ( JL. 0)
03 C5153 0605230000CC0012          ZQR   OR,DATO                             ; DATAOUT:= Q
04 C5154 0605730000231011          RZR   OR,ICS,LAD              CONT WRT    ; LAD:= ICS  , WRT
05 C5155 4631000002CC4016          CALL  F,F,CINTR                           ; CLEAR ALL INTERRUPTS
06 C5156 0655541740CC0100          XA    AND,CIOST,WRK5          CONT 0 1    ; IF CPUIOSTATUS(21:23)<>0 THEN
07 C5157 4631000000425241          JMP   T,NZ,SYFL4                          ;   GOTO SYSTEMFAULT4
08 C5160 4624731000CC0300          IZQ   OR,4000,14                          ; Q:= 8'40000014
09 C5161 4604013600CC0000          AGQ   SUNO,WRK2                           ; Q:= 40000000+DEV NO*10;
10 C5162 4631000002CC5207          CALL  F,F,WAUTO                           ; SEND AUTO(BASE,ADDRESS);
11 C5163 0655541740CC0100          XA    AND,CIOST,WRK5          CONT 0 1    ; IF CPUIOSTATUS(21:23)=0 THEN
12 C5164 4631000000425171          JMP   T,NZ,AUT01                          ;
13
14                          AUT00:                                           ; BEGIN
15 C5165 4605730000241011          RZR   OR,ICS,LAD              CONT READJ  ;   LAD:= ICS  , READ INSTRUCTION
16 C5166 4613531471CC2006          CABR  OR,.1B17,CNTR,CNTR,CNTR0            ;   RUN:= 1
17 C5167 4613551471CC1506          CABR  CAND,.4,CNTR,CNTR,CNTR0             ;
18 C5170 7405230000CC0000          ZQ    OR                      NEXT        ;   GOTO NEXT INSTR
19                                                                           ; END ELSE
20                          AUT01:                                           ;
21 C5171 4665623000CC1000          IQ    SUB0,4000,40                        ; IF Q<>DISC ADDR THEN
22 C5172 0631000000425243          JMP   T,NZ,SYFL5                          ;   GOTO SYSTEMFAULT5
23
24                          ;TRY THE SECOND IDA801 CONTROLLER
25 C5173 0624731000CC0200          IZQ   OR,4000,10                          ; Q:= 40000010;
26 C5174 4613730034CC1500          CZB   OR,.4,WRK2                          ; WRK2:= 4;
27 C5175 4631000002CC5207          CALL  F,F,WAUTO                           ; SEND AUTO(WRK2,Q);
28 C5176 0655541740CC0100          XA    AND,CIOST,WRK5          CONT 0 1    ; IF CPUIOSTATUS(21:23)=0 THEN
29 C5177 0631000000C25165          JMP   F,NZ,AUT00                          ;   GOTO AUT00;
```

```
10121 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;TRY THE FPA TRANSMITTER
03
04 C5200 4624731777777500          IZQ    OR,7777,7764                    ; Q:= 14- 3*10 , DEV BASE FOR FPA XM
05 C5201 0607230034000000          ZQB    OR,WRK2                         ; WRK2:= Q;
06 C5202 4624731000000600          IZQ    OR,4000,30                      ; Q:= 8'40000030, ADDR OF FPA XT;
07 C5203 4631000002005207          CALL   F,F,WAUTO                       ; SEND AUTO;
08 C5204 0655541740000100          XA     AND,CIOST,WRK5       CONT U 1   ; IF CPUIOSTATUS(21:23)<>0 THEN
09 C5205 0631000000425245          JMP    T,NZ,SYFL6                      ;   GOTO SYSTEMFAULT 6;
10 C5206 4631000000000020          JMP    F,F,IDLE                        ; GOTO IDLE;
11
12                          WAUTO:                                        ; SEND AUTOLOAD:
13                                                                        ; WRK2=DEVICE BASE, Q=ADDRESS;
14 C5207 4605431600000012          ZAR    OR,WRK2,DATO                    ; DATO:= DEVICE BASE;
15 C5210 4611730000233611          CZR    OR,.8,LAD            CONT WRT   ; LAD:= 8   , WRT
16 C5211 0655541740000100          XA     AND,CIOST,WRK5       CONT U 1   ; IF CPUIOSTATUS(21:23)<>0 THEN
17 C5212 4631000000425241          JMP    T,NZ,SYFL4                      ;   GOTO SYSTEMFAULT4
18 C5213 2005230000230011          ZQR    OR,LAD               RTN  WRT   ; LAD:= ADDRESS   , WRT
19
20                          ;BOOTSTRAPLOADER
21
22                          BOOTL:                                        ;ADDR  DEV DESCRIPTION   CHAN PROGRAM
23 C5214 6024731777660000          IZQ    OR,7777,3000         RTN        ; 12                     INITIALIZE
24 C5215 6024730000000240          IZQ    OR,0,12              RTN        ; 14   START OF CH PROG
25 C5216 6024730000000500          IZQ    OR,0,24              RTN        ; 16   STATUS ADDR
26 C5217 2024730000010000          IZQ    OR,0,400             RTN        ; 20   INTR DESTINATION  READ
27 C5220 2024730000000540          IZQ    OR,0,26              RTN        ; 22   INTR LEVEL        FIRST ADDR
28 C5221 6024730000030000          IZQ    OR,0,1400            RTN        ; 24                     CHAR COUNT
29 C5222 6024730000170000          IZQ    OR,0,7400            RTN        ; 26                     STOP
```

```
10122 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;GET AND PUT GENERAL REGISTER INSTRUCTIONS
03                          ;*****************************************
04
05                          ;GET GENERAL REGISTER, GG, OPCODE = 34
06
07                          GGE:                                                ;GGE:
08 C5223 0624730000C70020           IZQ   OR,0,3401                           ;
09 C5224 0631000002CC0031           CALL  F,F,ESCAP                           ; CALL ESCAPE(3401)
10                          GG:                                                 ;GG:
11 C5225 4635431335CC0060           SRZAB OR,SB,WRK3,Z,NL,F,F                 ; WRK3:=SB/2
12 C5226 4631000002C04214           CALL  F,F,GETRG                           ; WRK0:= GETREG(WRK3)
13 C5227 7407431500660000           ZAB   OR,WRK0,W               NEXT TSTIN  ; W:= WRK0, TEST INT
14                                                                            ; GOTO NEXT INSTR
15
16                          ;PUT GENERAL REGISTER, GP, OPCODE = 57, PRIVILEGED INSTRUCTION
17
18                          GPE:                                                ;GPE:
19 C5230 4624730000137020           IZQ   OR,0,5741                           ;
20 C5231 0631000002CC0031           CALL  F,F,ESCAP                           ; CALL ESCAPE(5741)
21                          GP:                                                 ;GP:
22 C5232 0631000000160032           JMP   F,MMODE,INT0                        ; IF -,MONMODE THEN GOTO INT0
23 C5233 0607430032CC0000           ZAB   OR,W,WRK0                           ; WRK0:= W
24 C5234 4635431335000060           SRZAB OR,SB,WRK3,Z,NL,F,F                 ; WRK3:= SB/2
25 C5235 0631000002CC4275           CALL  F,F,PUTRG                           ; PUTREG(WRK0,WRK3)
26 C5236 7405230000660000           ZQ    OR                      NEXT TSTIN  ; TEST INT, GOTO NEXT INST
27
28                          ;SYSTEMFAULT 3-6
29                          ;***************
30
31                          SYFL3:                                              ; SYSTEMFAULT CAUSE 3
32 C5237 0624730000CC0060           IZQ   OR,0,3                              ; ERROR IN FETCH OF FIRST INSTRUCTION
33 C5240 4631000000CC4021           JMP   F,F,SYFLT                           ; AT POWERRESTART
```

```
10123 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          SYFL4:                                          ; SYSTEMFAULT CAUSE 4
03 C5241 4624730000CC0100           IZQ   OR,0,4                          ; ERROR IN WRITING THE AUTOLOAD PROGRAM
04 C5242 4631000000CC4021           JMP   F,F,SYFLT                       ;
05
06                          SYFL5:                                          ; SYSTEMFAULT CAUSE 5
07 C5243 0624730000CC0120           IZQ   OR,0,5                          ; BUSERROR FROM AUTOLOADDEVICE,  FPAR
08 C5244 4631000000CC4021           JMP   F,F,SYFLT                       ;
09
10                          SYFL6:                                          ; SYSTEMFAULT CAUSE 6
11 C5245 0624730000CC0140           IZQ   OR,0,6                          ; BUSERROR FROM AUTOLOAD,  FPAX AND DISC
12 C5246 4631000000CC4021           JMP   F,F,SYFLT                       ;
13
14
15                          ;SYFLT:                                         ;SYSTEMFAULT:
16                          .YFLT:                                          ;
17 C5247 0607230037CC0000           ZQB   OR,WRK5                         ; WRK5:= Q,  CAUSE
18 C5250 0624700515312460           IZQ   OR,"S*16.+("Y/16.),"Y*256.+"S   ; TYPE 'SYSTEMFAULT<32>' ON TCP
19 C5251 4631000002CC4013           CALL  F,F,TPTXT                       ;
20 C5252 0624700521052320           IZQ   OR,"T*16.+("E/16.),"E*256.+"M   ;
21 C5253 4631000002CC4013           CALL  F,F,TPTXT                       ;
22 C5254 4624700431012520           IZQ   OR,"F*16.+("A/16.),"A*256.+"U   ;
23 C5255 4631000002CC4013           CALL  F,F,TPTXT                       ;
24 C5256 4624700461241000           IZQ   OR,"L*16.+("T/16.),"T*256.+32.  ;
25 C5257 4631000002CC4013           CALL  F,F,TPTXT                       ;
26 C5260 0604431740CC0000           ZAQ   OR,WRK5                         ; Q:= WRK5
27 C5261 4631000002CC4160           CALL  F,F,TPNUM                       ; TYPE NUMBER(CAUSE)
28 C5262 4613551471CC2006           CABR  CAND,.1B17,CNTR,CNTR,CNTRO      ; CLEAR RUN LAMPS
29 C5263 4613531471CC1506           CABR  OR,.4,CNTR,CNTR,CNTRO           ;
30                          SYFLR:                                          ; WHILE NOT CPUINT DO
31 C5264 0624730050CC0000           IZQ   OR,240,0                        ; BEGIN
32 C5265 0631000002CC4012           CALL  F,F,DELAY                       ;  DELAY(8'24000000);
33 C5266 4613551471CC3706           CABR  CAND,.1,CNTR,CNTR,CNTRO         ;  SET FAULT LAMP;
34 C5267 0624730050CC0000           IZQ   OR,240,0                        ;  DELAY(8'24000000);
35 C5270 0631000002CC4012           CALL  F,F,DELAY                       ;
36 C5271 4613551471CC3706           CABR  OR,.1,CNTR,CNTR,CNTRO           ;  CLEAR FAULT LAMP;
37 C5272 4631000000C6C0023          JMP   T,CPUIN,INTIN                   ; END;
38 C5273 0631000000CC5264           JMP   F,F,SYFLR                       ; GOTO CPUINTERRUPT;
```

```
10124 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;*************************************************
03                          ;
04                          ;       CPU823 CACHE TEST
05                          ;
06                          ;*************************************************
07
08                          ;PROCEDURE TO COUNT THE NUMBER OF CACHE MODULES
09                          ;EXIT:  WRK5= 1B22*IF SET 0 + 1B23*IF SET 1
10
11                          CAMC0:                                              ;COUNT CAMS
12 C5274 4605240000CC0024           ZQR   AND,CAMBC                             ; CAMBYPASSCONTR:=0
13 C5275 0607340037CC0000           ZBB   AND,WRK5                              ; WRK5:= 0
14 C5276 0653702036CC3700           CZB   ADDO,.1,WRK4                          ; FOR N:=0,1 DO
15 C5277 0624730000CC1200           IZQ   OR,0,50                               ; BEGIN
16 C5300 0605230000CC0041 CMC01:    ZQR   OR,CAMCN                              ;   CAMCONTROLREG:= IF N=0 THEN 8'50
17                                                                              ;     ELSE 8'30
18 C5301 0605240000230011           ZQR   AND,LAD               CONT WRT        ;   LAD:= 0 ,  WRT
19 C5302 4655730000CC0100           XZ    OR,CIOST              CONT 0 1        ;   C. SET VALID BIT
20 C5303 4605230000CC0040           ZQR   OR,CAMTD                              ;   CAMTESTDATA:= CAMCONTROLREG
21 C5304 4605240000210011           ZQR   AND,LAD               CONT READ       ;
22 C5305 0655730000CC4100           XZ    OR,IOSTA              CONT 0 1        ;   LAD:= 0  , READ
23 C5306 4611630000CC1441           CQR   OR,.3,CAMCN                           ;   CAMCONTR:=CAMCONTR OR 3
24 C5307 4605240000210011           ZQR   AND,LAD               CONT READ       ;   LAD:= 0  , READ
25 C5310 4657730033CC4100           XZB   OR,IOSTA,WRK1         CONT 0 1        ;   WRK1:= IOSTATUS
26 C5311 4607310036CC0000           ZBB   SUN,WRK4                              ;   IF HIT(N) THEN
27 C5312 0631000000C25320           JMP   F,NZ,CMC02                            ;     WRK5(22+N):= 1
28 C5313 0624730000CC0600           IZQ   OR,0,30                               ;
29 C5314 4611541540CC2000           CA    AND,.1B17,WRK1                        ;
30 C5315 0631000000425300           JMP   T,NZ,CMC01                            ;
31 C5316 4653702037CC3700           CZB   ADDO,.1,WRK5                          ;
32 C5317 0631000000CC5300           JMP   F,F,CMC01                             ;
33 C5320 4611541540CC1600 CMC02:    CA    AND,.1B19,WRK1                        ;
34 C5321 2031000000C25322           JMP   F,NZ,.+3              RTN             ;
35 C5322 2013531777CC3700           CAB   OR,.1,WRK5,WRK5       RTN             ; END
```

```
10125 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;PROCEDURE TO CLEAR THE CACHE MEMORY MODULES
03
04 05323 0653702036CC3700 CAMCL: CZB    ADDO,.1,WRK4                          ; FOR N:= 0,1 DO
05 05324 4624730000001000        IZQ    OR,0,40                               ; BEGIN
06 05325 0605240000000040        ZQR    AND,CAMTD                             ;   CAMTD:= 0
07 05326 0605141437CC0000        AB     AND,.2,WRK5                           ;   IF SET N PRESENT THEN
08 05327 0631000000025337        JMP    F,NZ,CAMC2                            ;   BEGIN
09 05330 0605230000000041 CAMC1: ZQR    OR,CAMCN                              ;     CAMCN:= IF N=0 THEN 8'40 ELSE 8'20
10 05331 4605240000000024        ZQR    AND,CAMBC                             ;     CAMBC:=0
11 05332 4607340035CC0000        ZBB    AND,WRK3                              ;     FOR WRK3:= 0 STEP 2 UNTIL 8'20000 DO
12 05333 4251712000000304        CZR    SUNO,B12,LCD           PUSH           ;
13 05334 0605431640210011        ZAR    OR,WRK3,LAD            CONT READ      ;       LAD:= WRK3  , READ
14 05335 0655730000004100        XZ     OR,IOSTA               CONT 0 1       ;
15 05336 6637101435114060        NSABB  ADD,.2,WRK3,NL,F,LCN   LRTN           ;   END
16 05337 4607310036CC0000 CAMC2: ZBB    SUN,WRK4                              ;
17 05340 4631000000025344        JMP    F,NZ,CAMC3                            ;
18 05341 4624730000000400        IZQ    OR,0,20                               ;
19 05342 0611541740003700        CA     AND,.1,WRK5                           ;
20 05343 0631000000425330        JMP    T,NZ,CAMC1                            ; END
21                          CAMC3:                                            ;CAMC3:  C. INIT 'READY' AFTER TESTWRITE
22 05344 0611730000001241        CZR    OR,.11,CAMCN                          ; CAMCN:= 8'13
23 05345 4605240000210011        ZQR    AND,LAD                CONT READ      ; LAD:= 0   , READ
24 05346 6055730000004100        XZ     OR,IOSTA               RTN 0 1        ;
25
26                          ;INIT CACHE MODULES
27
28 05347 4604730000005000 CAMIN: RZQ    OR,SWTCH                              ; IF -,SWITCH(-,CAM AVAILABLE) THEN
29 05350 0625640000000040        IQ     AND,0,2                               ; BEGIN
30 05351 2031000002025274        CALL   F,NZ,CAMCO             RTN            ;   COUNT CAM
31 05352 0631000002005323        CALL   F,F,CAMCL                             ;   CLEAR CAM
32 05353 4605240000000024        ZQR    AND,CAMBC                             ;   CAM BYPASS CONT:= 0
33 05354 4611730000000741        CZR    OR,.15,CAMCN                          ;   CAMCONTROLREG:= 8'17
34 05355 0605240000230011        ZQR    AND,LAD                CONT WRT       ;   LAD:= 0 ,  WRT
35 05356 2055730000000100        XZ     OR,CIOST               RTN 0 1        ;   C. SET VALID BIT
36                                                                            ; END
37
```

```
10126 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;CAMER:                                                ;CAMERROR
03                          .AMER:                                                 ;
04 C5357 4631000002CC5347           CALL   F,F,CAMIN                               ; INIT CAM
05 C5360 0611541.440CC2000          CA     AND,.1B17,CNTR                          ; IF CNTR.17 =0 THEN
06 C5361 0631000000C20020           JMP    F,NZ,IDLE                               ; GOTO IDLE
07 C5362 0611541.440CC1500          CA     AND,.4,CNTR                             ; IF CNTR.21 =1 THEN
08 C5363 0631000000422001           JMP    T,NZ,DISPA                              ; GOTO DISPATCHING POINT
09 C5364 4605730000241011           RZR    OR,ICS,LAD              CONT READJ      ; LAD:= ICS  , READ INSTRUCTION
10 C5365 7545230000CC0000           ZQ     OR                      NEXT 0 3        ; GOTO NEXT INSTRUCTION
11
12                          ;CAMTEST:                                              ;CAMTEST START
13                          .AMTEST:                                               ;
14 C5366 4624730000CC0040           IZQ    OR,0,2                                  ; IF CAM NOT AVAILABLE THEN
15 C5367 0605640000CC5000           RQ     AND,SWTCH                               ;   GOTO TEST CONTINUE
16 C5370 0631000000426005           JMP    T,NZ,T0.4                               ;
17 C5371 4613531.471CC3406          CABR   OR,.7,CNTR,CNTR,CNTRO                   ;
18 C5372 4613551.471CC3706          CABR   CAND,.1,CNTR,CNTR,CNTRO                 ; TURN ERRORLAMP 3 ON
19 C5373 4624730000CC0160           IZQ    OR,0,7                                  ; IF TCP THEN
20 C5374 0631000002164013           CALL   F,MMODE,TPTXT                           ;   TYPE BELL
21 C5375 0631000002CC5274           CALL   F,F,CAMCO                               ; COUNT CAM
22 C5376 0631000002CC6004           CALL   F,F,SIZEM                               ; SIZE AND CLEAR MEM
23 C5377 0653702021CC3700           CZB    ADD0,.1,W1                              ; FOR SET:= 0,1 DO
24 C5400 0613730022CC2000           CZB    OR,.1B17,W2                             ; BEGIN
25 C5401 0624730000CC1200           IZQ    OR,0,50                                 ;   MASK:= IF SET=0 THEN 1B17 ELSE 1B19
26 C5402 0605141.437CC0000          AB     AND,.2,WRK5                             ;   IF SET PRESENT THEN
27 C5403 0631000000C25471           JMP    F,NZ,CEX3                               ;   BEGIN
28                          C04:                                                   ;
29 C5404 4607230036CC0041           ZQBR   OR,WRK4,CAMCN                           ;     CAMCONTROL:=IF SET=0 THEN 8'50
30 C5405 0607340020CC0000           ZBB    AND,W0                                  ;     ELSE 8'30
31                          C05:                                                   ;     FOR STARTWINDOW:= 0 STEP 8'10000 UN
32 C5406 0607431035CC0003           ZABR   OR,W0,WRK3,1CD                          ;     BEGIN
33 C5407 0605431700CC0041           ZAR    OR,WRK4,CAMCN                           ;       FOR WRK:= 0 STEP 2 UNTIL 8'10000 D
34 C5410 4251712000CC0304           CZR    SUN0,B12,LCD            PUSH            ;       BEGIN
35 C5411 4645113035CC0040           ABR    SUN0,W0,WRK3,CAMTD                      ;         CAMTD:= WRK
36 C5412 0605431640210011           ZAR    OR,WRK3,LAD             CONT READ       ;         LAD:= STARTW+WRK  , READ
37 C5413 0655730000CC4100           XZ     OR,IOSTA                CONT 0 1        ;         C. WRITE TESTDATA
38 C5414 6637101.435114060          NSABB  ADD,.2,WRK3,NL,F,LCN    LRTN            ;       END FOR WRK
39 C5415 4611531700001441           CAR    OR,.3,WRK4,CAMCN                        ;       CAMCONTROL:= CAMCONTROL OR 3
```

```
10127 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                                                                          ;       FOR IC:= W0,WU+2,...,LAST,0,...,WU-
03                          C10:                                            ;       DO BEGIN
04 C5416 4213551224CC3305           CABR  CAND,.1B1,STAT,STAT,CPUST PUSH    ;          CLEAR ERROR FLAG, SETUP LOOP
05 C5417 4644523000CC1000           RAQ   SUBO,ICS,WU                       ;          IF IC< STARTWINDOW THEN
06 C5420 0631000000C15427           JMP   F,NEG,C11                         ;          BEGIN
07 C5421 4607730034211011           RZBR  OR,ICS,WRK2,LAD         CONT READ ;             LAD:=WRK2:=IC , READ
08 C5422 4657541133CC4100           XAB   AND,IOSTA,W2,WRK1       CONT U 1  ;             STATUS:= MASK AND IOSTATUS
09 C5423 4631000002C25526           CALL  F,NZ,CMHER                        ;             IF STATUS=U THEN CAMHITERROR
10 C5424 4655523600CC4000           XA    SUBO,DATI,WRK2                    ;             IF DATAIN<> IC THEN
11 C5425 0631000002425531           CALL  T,NZ,CMRER                        ;               CAMREADERROR
12 C5426 4631000000CCC5460          JMP   F,F,CEX1                          ;          END ELSE
13 C5427 4664612004C00000 C11:      IOQ   SUNO,2,U                          ;          IF IC< STARTWINDOW + 8'20000 THE
14 C5430 0631000000015450           JMP   F,NEG,C12                         ;          BEGIN
15 C5431 4607730034211011           RZBR  OR,ICS,WRK2,LAD         CONT READ ;             LAD:=WRK2:= IC ,  READ
16 C5432 4657541133CC4100           XAB   AND,IOSTA,W2,WRK1       CONT U 1  ;             STATUS:= MASK AND IOSTATUS
17 C5433 0631000002425526           CALL  T,NZ,CMHER                        ;             IF STATUS<>U THEN CAMHITERROR
18 C5434 4605730000CC1012           RZR   OR,ICS,DATO                       ;             DATAOUT:= IC
19 C5435 0605730000231011           RZR   OR,ICS,LAD              CONT WRT  ;             LAD:= IC  , WRT
20 C5436 0647523034CC1000           RAB   SUBO,ICS,WU,WRK2                  ;             IF DATI<> IC-STARTWINDOW THEN
21 C5437 4655523600CC4000           XA    SUBO,DATI,WRK2                    ;               CAMREADERROR
22 C5440 0631000002425531           CALL  T,NZ,CMRER                        ;
23 C5441 4655730000CC0100           XZ    OR,CIOST                CONT U 1  ;
24 C5442 4611730000CC3724           CZR   OR,.1,CAMBC                       ;             CAMBYPASSREG:= 1
25 C5443 4607730034211011           RZBR  OR,ICS,WRK2,LAD         CONT READ ;             LAD:=WRK2:= IC  , READ
26 C5444 4715523600CC4000           XA    SUBO,DATI,WRK2          CONT U 2  ;             IF DATI<>IC THEN
27 C5445 0631000002425534           CALL  T,NZ,CMWER                        ;               CAMWRITEERROR
28 C5446 4605240000CC0024           ZOR   AND,CAMBC                         ;             CAMBYPASSREG:= U
29 C5447 4631000000CC5460           JMP   F,F,CEX1                          ;          END ELSE
30 C5450 0645513140CC1000 C12:      RA    SUNO,ICS,W3                       ;
31 C5451 0631000000415524           JMP   T,NEG,C50                         ;          BEGIN
32 C5452 4607730034211011           RZBR  OR,ICS,WRK2,LAD         CONT READ ;             LAD:=WRK2:= IC ,  READ
33 C5453 4657541133CC4100           XAB   AND,IOSTA,W2,WRK1       CONT U 1  ;             STATUS:= MASK AND IOSTATUS
34 C5454 4631000002C25526           CALL  F,NZ,CMHER                        ;             IF STATUS=U THEN CAMHITERROR
35 C5455 0607340034CC0000           ZBB   AND,WRK2                          ;             WRK2:= U
36 C5456 0615730000CC4000           XZ    OR,DATI                           ;             IF DATI<>U THEN
37 C5457 0631000002425531           CALL  T,NZ,CMRER                        ;               CAMREADERROR
```

```

                                      CEX1:                                            ;          END
C5460 4631000002006003          CALL  F,F,LOOP                                         ;          TEST LOOPING
C5461 4045501400001003          RAR   ADD,ICS,.2,ICD             POP                   ;
C5462 0645523000001000  C13:    RA    SUBO,ICS,WO                                      ;
C5463 0631000000425416          JMP   T,NZ,C10                                         ;        END FOR IC
C5464 4624730000400000          IZQ   OR,2,0                                           ;
C5465 0607001020000000          AQB   ADD,WO,WO                                        ;
C5466 0605123160000000          AB    SUBO,W3,WO                                       ;      END FOR STARTWINDOW
C5467 0631000000015406          JMP   F,NEG,C05                                        ;
C5470 0631000002006004          CALL  F,F,SIZEM                                        ;      SIZE AND CLEAR MEM
                        CEX3:                                                          ;    END IF SET PRESENT
C5471 4607310021000000          ZBB   SUN,W1                                           ;  END FOR SET
C5472 0631000000025477          JMP   F,NZ,C40                                         ;
C5473 0624730000000600          IZQ   OR,0,30                                          ;
C5474 0613730022001600          CZB   OR,.1B19,W2                                      ;
C5475 0611541740003700          CA    AND,.1,WRK5                                      ;
C5476 0631000000425404          JMP   T,NZ,C04                                         ;
                        C40:                                                           ;
C5477 4651702000001341          CZR   ADDO,.10,CAMCN                                   ;  CAMCONTROLREG:= 8'13
C5500 4651523740001400          CA    SUBO,.3,WRK5                                     ;  IF SET 0 AND 1 PRESENT THEN
C5501 0631000000426005          JMP   T,NZ,10.4                                        ;  BEGIN  C. TEST RANDOMGENERATOR
C5502 4213551224003305          CABR  CAND,.1B1,STAT,STAT,CPUS1 PUSH                   ;    CLEAR ERRORFLAG , SETUP LOOP
C5503 4605240000210011          ZQR   AND,LAD                    CONT READ             ;    LAD:= 0  , READ
C5504 0655730000004100          XZ    OR,IOSTA                   CONT 0 1              ;
C5505 4624730000400000          IZQ   OR,2,0                                           ;
C5506 4607230035210011          ZQBR  OR,WRK3,LAD                CONT READ             ;    WRK3:=LAD:= 8'20000  , READ
C5507 0655730000004100          XZ    OR,IOSTA                   CONT 0 1              ;
C5510 0624730000002400          IZQ   OR,0,120                                         ;    Q:=MASK:= 1BIT17+1B19
C5511 4605240000210011          ZQR   AND,LAD                    CONT READ             ;    LAD:= 0  , READ
C5512 4657640033004100          XQB   AND,IOSTA,WRK1             CONT 0 1              ;    WRK1:= MASK AND IOSTATUS
C5513 0605431640210011          ZAR   OR,WRK3,LAD                CONT READ             ;    LAD:= 8'20000  , READ
C5514 0657640034004100          XQB   AND,IOSTA,WRK2             CONT 0 1              ;    WRK2:= IOSTATUS AND MASK
C5515 4653702032001500          CZB   ADDO,.4,WRK0                                     ;    WRK0:= 5
C5516 0604161574000000          ABQ   EXOR,WRK1,WRK2                                   ;    Q:= WRK1 EXOR WRK2
C5517 4665622000002400          IQ    SUBO,0,120                                       ;    IF Q<> MASK THEN
C5520 0631000002426002          CALL  T,NZ,TERROR                                      ;      TERROR
C5521 4631000002006003          CALL  F,F,LOOP                                         ;    TEST LOOPING
```

```
10129 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 C5522 0005230000000000          ZQ     OR                    POP          ; END IF SET 0 AND 1
03                                                                            ; CAMTEST FINISHED
04 C5523 0631000000006005          JMP    F,F,T0.4                           ; GOTO TEST REPEATTEST
05
06                              C50:                                         ;C50:
07 C5524 4005240000000003          ZQR    AND,ICD               POP          ;
08 C5525 0631000000005462          JMP    F,F,C13                            ;
09
10                              CMHER:                                       ;CAMHITERROR:
11 C5526 0607431432000000          ZAB    OR,.2,WRK0                         ; WRK0:= 2
12 C5527 0607431134000000          ZAB    OR,W2,WRK2                         ; WRK2:= MASK
13 C5530 4631000000006002          JMP    F,F,TERROR                         ; GOTO TERROR
14
15                              CMRER:                                       ;CAMREADERROR:
16 C5531 0613730032001400          CZB    OR,.3,WRK0                         ; WRK0:= 3
17 C5532 4617730033004000          XZB    OR,DATI,WRK1                       ; WRK1:= DATI
18 C5533 4631000000006002          JMP    F,F,TERROR                         ; GOTO TERROR
19
20                              CMWER:                                       ;CAMWRITEERROR:
21 C5534 4613730032001500          CZB    OR,.4,WRK0                         ; WRK0:= 4
22 C5535 4617730033004000          XZB    OR,DATI,WRK1                       ; WRK1:= DATI
23 C5536 4631000000006002          JMP    F,F,TERROR                         ;
```

```
10130 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          FILL 60CC       ;FILL SEGMENT WITH ONES
03                          .EOT
04
05                          ;FILE: T4, EDITED: 850124/LBJ
06
07                          ;*************************************************
08                          ;
09                          ;          CPU 823 MICRO TEST
10                          ;
11                          ;*************************************************
12
13                    CC0001 .NOMAC 1
14                    C22000 .LOC 60CC*3
15 C6CC0 0631000000CCC6006          JMP   F,F,.TX                          ; GOTO TX
16 C6001 4631000000CCC6062          JMP   F,F,.TEST                        ; GOTO ATEST
17 C6002 4631000000CCC6111          JMP   F,F,.ERROR                       ; GOTO TERROR
18 C6003 0631000000CCC6104          JMP   F,F,.LOOP                        ; GOTO LOOP
19 C6004 4631000000CCC7656          JMP   F,F,.IZEM                        ; GOTO SIZEMEM
20 C6005 0631000000CCC6077          JMP   F,F,.T0.4                        ; GOTO T0.4
21
22                          ;TX:                                           ;
23                          .TX:                                           ;TX:
24 C6006 0631000002CCC4014          CALL  F,F,GINUM                        ; WRK2:= TEST NO
25 C6007 4624730000140260           IZQ   OR,0,TCASE/3                     ; Q:= ADDR(TCASE);
26 C6010 4651513600CC3600           CA    SUN0,,8,WRK2                     ; IF WRK2<8 THEN
27 C6011 0631001601410000           JXAQ  ADD,WRK2,T,NEG                   ;   GOTO Q+WRK2
28 C6012 4631000000CCC4005          JMP   F,F,TCPER                        ; ELSE
29                                                                         ;   GOTO TCP ERROR
30
31                          TCASE:                                         ;
32 C6013 4631000000CCC6023          JMP   F,F,T0                           ; CPUTEST, MEMTEST NORMAL MODE, CAMTEST
33 C6014 4631000000CCC6026          JMP   F,F,T1                           ; CPUTEST, MEMTEST SHORT MODE, CAMTEST
34 C6015 4631000000CCC6031          JMP   F,F,T2                           ; CPUTEST
35 C6016 4631000000CCC6034          JMP   F,F,T3                           ; MEMTEST NORMAL MODE
36 C6017 4631000000CCC6037          JMP   F,F,T4                           ; MEMTEST SHORT MODE
37 C6020 0631000000CCC6042          JMP   F,F,T5                           ; CAMTEST
38 C6021 4631000000CCC6045          JMP   F,F,T6                           ; SIZE AND CLEAR MEMORY
39 C6022 4631000000CCC6051          JMP   F,F,T7                           ; LOOP TO ADJUST BUSINTERFACE
```

```
10131 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          T0:                                     ;T0:  C. CPUTEST, MEMTEST NORMAL MODE
03 C6023 062473017000000            IZQ   OR,0740,0                 ;     CAMTEST
04 C6024 460723002400005            ZQBR  OR,STAT,CPUST             ;
05 C6025 063100000006071            JMP   F,F,T0.1                  ;
06
07                          T1:                                     ;T1:  C. CPUTEST, MEMTEST SHORT MODE
08 C6026 462473007000000            IZQ   OR,0340,0                 ;        CAMTEST
09 C6027 460723002400005            ZQBR  OR,STAT,CPUST             ;
10 C6030 063100000006071            JMP   F,F,T0.1                  ;
11
12                          T2:                                     ;T2:  C. CPUTEST
13 C6031 462473004000000            IZQ   OR,0200,0                 ;
14 C6032 460723002400005            ZQBR  OR,STAT,CPUST             ;
15 C6033 063100000006071            JMP   F,F,T0.1                  ;
16
17                          T3:                                     ;T3:  C. MEMTEST NORMAL MODE
18 C6034 062473012000000            IZQ   OR,0500,0                 ;
19 C6035 460723002400005            ZQBR  OR,STAT,CPUST             ;
20 C6036 463100000006073            JMP   F,F,T0.2                  ;
21
22                          T4:                                     ;T4:  C. MEMTEST SHORT MODE
23 C6037 462473002000000            IZQ   OR,0100,0                 ;
24 C6040 460723002400005            ZQBR  OR,STAT,CPUST             ;
25 C6041 463100000006073            JMP   F,F,T0.2                  ;
26
27                          T5:                                     ;T5:  C. CAMTEST
28 C6042 462473001000000            IZQ   OR,0040,0                 ;
29 C6043 460723002400005            ZQBR  OR,STAT,CPUST             ;
30 C6044 463100000006075            JMP   F,F,T0.3                  ;
31
32                          T6:                                     ;T6:  C. SIZEMEM
33 C6045 463100000205044            CALL  F,F,INITR                 ; INIT REGISTERS
34 C6046 063100000206004            CALL  F,F,SIZEMEM               ; SIZE AND CLEAR MEM
35 C6047 063100000014005            JMP   F,NEG,ICPER               ; IF IOERROR THEN ICPER
36 C6050 063100000004004            JMP   F,F,SINGL                 ; GOTO SINGLE INSTRUCTION
```

```
10132 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          T7:                                                 ;T7:  C. LOOP TO ADJUST DESKEW DELAY
03 C6051 461473000C0C0100          XZQ    OR,CIOSTA                           ;
04 C6052 462464007GCC0000          1QQ    AND,0370,0                          ;
05 C6053 061363003200C300          CQB    OR,.1B12,WRKU                       ;
06 C6054 466273153200C500          HZB    OR,SWAP,WRKU,WRKU                   ; WRKU:= 1BU + CPUADDR
07 C6055 460524000C0C0012          ZQR    AND,DATO                            ; DATAOUT:= U
08 C6056 421353122400C3305         CABR   OR,.1B1,STAT,STAT,CPUST  PUSH       ; REPEAT
09 C6057 060543150023U011          ZAR    OR,WRKU,LAD              CONT WRT   ;   LAD:= WRKU  , WRT
10 C6060 065473000C0C0100          XZQ    OR,CIOST                 CONT U 1   ;   Q:= CPUIOSTATUS
11 C6061 463100000200G003          CALL   F,F,LOOP                           ; UNTIL CPUINTERRUPT
12
13                          .TEST:                                             ;
14                          ;ATEST:                                            ;AUTOLOAD ENTRY:
15 C6062 462473117GCC0000          IZQ    OR,4740,0                           ; IF SWITCH 5 = U THEN
16 C6063 460723002400C005          ZQBR   OR,STAT,CPUST                       ;   CPUTEST, SHORT MEM, CAMTEST
17 C6064 460473000C0C5000          RZQ    OR,SWTCH                            ; ELSE
18 C6065 062564000C0C1000          1Q     AND,U,U040                          ;   CPUTEST, LONG MEM, CAMTEST
19 C6066 063100000G426071          JMP    T,NZ,TU.1                           ;
20 C6067 062473107GCC0000          IZQ    OR,4340,0                           ;
21 C6070 460723002400C005          ZQBR   OR,STAT,CPUST                       ;
22
23                                                                            ; REPEAT C.TEST CPUTEST
24 C6071 061154120GCC2700 TU.1:    CA     AND,.1B4,STAT                       ;   IF STAT(4) THEN GOTO CPUTEST
25 C6072 063100000G426173          JMP    T,NZ,CPUTEST                        ;
26                                                                            ;   C. TEST MEMTEST
27 C6073 461154120GCC2600 TU.2:    CA     AND,.1B5,STAT                       ;   IF STAT(5) THEN GOTO MEMTEST
28 C6074 463100000G427475          JMP    T,NZ,MEMTEST                        ;
29                                                                            ;   C. TEST CAMTEST
30 C6075 461154120GCC2500 TU.3:    CA     AND,.1B6,STAT                       ;   IF STAT(6) THEN GOTO CAMTEST
31 C6076 463100000G424017          JMP    T,NZ,CAMTEST                        ;
32                          .TU.4:                                             ;
33                          ;TU.4:                                             ;
34 C6077 463100000G6C0023          JMP    T,CPUIN,INTIN                       ;   IF CPUINTERRUPT THEN GOTO INTIN
35 C6100 461154120GC03100          CA     AND,.1B3,STAT                       ;   IF NORMAL MODE THEN
36 C6101 063100000G426071          JMP    T,NZ,TU.1                           ;     REPEAT TEST
37 C6102 063100000G564015          JMP    T,MMODE,RAUTO                       ;   IF AUTOLOADENTRY THEN GOTO RAUTO
38 C6103 063100000C0C6071          JMP    F,F,TU.1                            ; UNTIL FALSE
```

```
10133 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;CONTENTS OF REGISTER STAT AND CPUST
03                          ;  AUTOLOADENTRY       BIT 0
04                          ;  ERROR-AND LOOPFLAG  BIT 1
05                          ;  FULL TEST           BIT 3
06                          ;  CPUTEST             BIT 4
07                          ;  MEMTEST             BIT 5
08                          ;  CAMTEST             BIT 6
09                          ;  DOWNFLAG            BIT 7
10
11                          ;LOOP:                                          ;TEST LOOPING
12                          .LOOP:                                          ;
13                                                                          ; IF -,LOOPMODE THEN
14 C6104 4031000000176106           JMP   F,EMODE,LOOP1          POP        ;   IF CPUINTERRUPT THEN GOTOINTIN
15 C6105 6631000006C6C107           JMP   T,CPUIN,INT.2          LRTN       ; ELSE
16 C6106 6031000006C6C107 LOOP1:    JMP   T,CPUIN,INT.2          RTN        ; BEGIN
17
18 C6107 0653702030CC3700 INT.2:    CZB   ADDO,.1,.2                        ;   POP
19 C6110 4631000000CC0023           JMP   F,F,INTIN                         ;   IF CPUINTERRUPT THEN GOTO INT.2
20                                                                          ;   ELSE LOOPRETURN
21                                                                          ; END
22
23                          ;TERROR:                                        ;ERRORROUTINE
24                          .ERROR:                                         ;
25 C6111 4631000C0566153            JMP   T,MMODE,ERRAUTO                   ; IF AUTOLOADENTRY THEN GOTO ERRAUTO
26 C6112 6031000000176113           JMP   F,EMODE,.+3            RTN        ; IF ERRORFLAG THEN RETURN
27 C6113 4641421500CC1202           ZAS   SUB,WRK0,12                       ; SAVE WRK0..2 AND Q
28 C6114 4641421540CC1302           ZAS   SUB,WRK1,13                       ;
29 C6115 4641421600CC1402           ZAS   SUB,WRK2,14                       ;
30 C6116 4641220000CC1502           ZQS   SUB,15                            ;
31 C6117 0653702030CC3700           CZB   ADDO,.1,.2                        ; .2:= 2
32 C6120 4624730425222440           IZQ   OR,2125,1122                      ;
33 C6121 4631000002CC4013           CALL  F,F,TPTXT                         ; TYPE 'ERR' ON TCP
34 C6122 4631000002CC4007           CALL  F,F,CRLF                          ; TYPE 'CR''LF'
35 C6123 0600730000CC1200           SZQ   OR,12                             ;
36 C6124 0631000002CC4006           CALL  F,F,TPOCT                         ; TYPE WRK0
37 C6125 4631000002CC4007           CALL  F,F,CRLF                          ;
38 C6126 4600730000CC1300           SZQ   OR,13                             ;
```

```
10134 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 C6127 0631000002CC4006          CALL  F,F,TPOCT                    ; TYPE WRK1
03 C6130 4631000002CC4007          CALL  F,F,CRLF                     ;
04 C6131 0600730000CC1400          SZQ   OR,14                        ;
05 C6132 0631000002CC4006          CALL  F,F,TPOCT                    ; TYPE WRK2
06 C6133 4631000002CC4007          CALL  F,F,CRLF                     ;
07 C6134 0604730000CC1000          RZQ   OR,ICS                       ;
08 C6135 0631000002CC4006          CALL  F,F,TPOCT                    ; TYPE IC
09 C6136 4631000002CC4007          CALL  F,F,CRLF                     ;
10 C6137 4631000002CC4010          CALL  F,F,GCHAR                    ; Q:= GETCHAR
11 C6140 4665622000CC2400          IQ    SUB0,0,"P                    ; IF Q='P' THEN
12 C6141 4631000000C26146          JMP   F,NZ,PROC                    ;   GOTO PROCEED
13 C6142 4665622000CC2340          IQ    SUB0,0,"N                    ; IF Q='N'
14 C6143 4031000000C26145          JMP   F,NZ,NLOOP           POP     ;   THEN GOTO NLOOP
15 C6144 4631000000CC4011          JMP   F,F,TCPI2                    ; GOTO AFTER TCP INPUT
16
17 C6145 0613531224CC3305 NLOOP:   CABR  OR,.1B1,STAT,STAT,CPUST      ; SET ERRORFLAG
18 C6146 4631000002CC4007 PROC:    CALL  F,F,CRLF                     ; TYPE 'CR''LF'
19 C6147 4603730032CC1200          SZB   OR,12,WRK0                   ; RESTORE WRK0
20 C6150 4603730033CC1300          SZB   OR,13,WRK1                   ; RESTORE WRK1
21 C6151 4603730034CC1400          SZB   OR,14,WRK2                   ; RESTORE WRK2
22 C6152 2000730000CC1500          SZQ   OR,15                RTN     ; RESTORE Q
23
24                          ERRAUTO:                                  ;ERRORPROCEDURE FOR AUTOLOADENTRY
25 C6153 0624730050CC0000          IZQ   OR,240,0                     ; REPEAT
26 C6154 0631000002CC4012          CALL  F,F,DELAY                    ;   DELAY(0.26 SEC)
27 C6155 0613551471CC1706          CABR  CAND,.1B18,CNTR,CNTR,CNTR0   ;   CLEAR AUTOLOAD LAMP
28 C6156 0624730050CC0000          IZQ   OR,240,0                     ;
29 C6157 0631000002CC4012          CALL  F,F,DELAY                    ;   DELAY(0.26 SEC)
30 C6160 0631000000C6107           JMP   1,CPUIN,INT.2                ;   IF CPUINTERRUPT THEN GOTO INT.2
31 C6161 0613531471CC1706          CABR  OR,.1B18,CNTR,CNTR,CNTR0     ;   SET AUTOLOAD LAMP
32 C6162 4631000000CC6153          JMP   F,F,ERRAUTO                  ; UNTIL FALSE
33
34                          TBERR:                                    ;BUSERROR
35 C6163 4653710032CC3700          CZB   SUN,.1,WRK0                  ; WRK0:= -2
36 C6164 0617730033CC4100          XZB   OR,IOSTA,WRK1                ; WRK1:= IOSTATUS
37 C6165 0613541573CC3400          CAB   AND,.7,WRK1,WRK1             ; WRK1:= IOSTATUS(21:23)
38 C6166 4631000000CC6002          JMP   F,F,TERROR                   ; GOTO TERROR
```

```
10135 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;ADM OF NUMBER OF CPUTEST RUNS
03
04 06167 0640720006001700 CPULP: SZQ    SUB,17                          ; IF RUNCOUNTER-1=0 THEN
05 06170 0631000000026073        JMP    F,NZ,T0.2                       ;   GOTO TEST MEMTEST
06 06171 0641220006001702        ZQS    SUB,17                          ; RUN COUNTER:= RUN COUNTER -1
07 06172 0631000000006201        JMP    F,F,B20                         ; REPEAT CPUTEST
08
09                          CPUTEST:                                       ;CPUTEST START
10 06173 4613531471003406        CABR   OR,.7,CNTR,CNTR,CNTR0           ; TURN LAMP 1 ON
11 06174 4613551471001506        CABR   CAND,.1B21,CNTR,CNTR,CNTR0      ;
12 06175 4624730000056700        IZQ    OR,0,2734                       ; RUN COUNTER:= 1500
13 06176 0641220006001702        ZQS    SUB,17                          ;
14 06177 4624730000000160        IZQ    OR,0,7                          ; IF TCP THEN
15 06200 0631000002164013        CALL   F,MMODE,TPTXT                   ;   TYPE(BELL)
16
17                          ;TEST OF THE LOAD IMMEDIATE FUNCTION
18
19                          ;THE IMMEDIATE OPERAND AND THE ALU'S ABILITY TO SHIFT LEFT
20                          ;ARE TESTED.
21                          ;THE IMOP IS LOADED WITH A SHIFTING 'ZERO' , WRK2 IS LOADED WITH THE
22                          ;EXPECTED DATA
23
24                          B20:                                           ;
25 06201 0613730032003600        CZB    OR,.8,WRK0                      ; WRK0:= 8'10
26 06202 0624730000144340        IZQ    OR,0,IMBASE/3-1                 ; Q:= ADDR(IMBASE)-1
27 06203 0607230037000000        ZQB    OR,WRK5                         ; WRK5:= POINTER TO IMOP TABLE
28 06204 0607121634000000        ABB    SUB,WRK2,WRK2                   ; WRK2:= -1
29                          B30:                                           ; REPEAT
30 06205 0607302037000000        ZBB    ADD0,WRK5                       ;   WRK5:= WRK5+1
31 06206 4213551224003305        CABR   CAND,.1B1,STAT,STAT,CPUS1  PUSH ;   CLEAR ERRORFLG  ,  PUSH
32 06207 0631431743000000        CXZA   OR,WRK5,F,F                     ;   Q:= DATAPATTERN
33 06210 4607230033000000        ZQB    OR,WRK1                         ;   WRK1:= Q
34 06211 0605023600000000        AQ     SUB0,WRK2                       ;   IF WRK2<>Q THEN
35 06212 0631000002426002        CALL   T,NZ,TERROR                     ;     TERROR
36 06213 4631000002006003        CALL   F,F,LOOP                        ;   TEST LOOPING
37 06214 4037330034000060        SLZBB  OR,WRK2,Z,NL,F,F        POP     ;   WRK2:= WRK2 SHIFT 1
38 06215 0607302032000000        ZBB    ADD0,WRK0                       ;   WRK0:= WRK0+1
39 06216 4631000000006205        JMP    F,F,B30                         ; UNTIL FALSE
```

```
10136 CPU823 MICROPROGRAM LBJ 850816 VERSION 8

                                IMBASE:                                              ;
06217 6024731777777760                  IZQ    OR,7777,7777          RTN          ;10
06220 2024731777777740                  IZQ    OR,7777,7776          RTN          ;11
06221 6024731777777700                  IZQ    OR,7777,7774          RTN          ;12
06222 2024731777777600                  IZQ    OR,7777,7770          RTN          ;13
06223 6024731777777400                  IZQ    OR,7777,7760          RTN          ;14
06224 2024731777777000                  IZQ    OR,7777,7740          RTN          ;15
06225 6024731777776000                  IZQ    OR,7777,7700          RTN          ;16
06226 2024731777774000                  IZQ    OR,7777,7600          RTN          ;17
06227 6024731777770000                  IZQ    OR,7777,7400          RTN          ;20
06230 2024731777760000                  IZQ    OR,7777,7000          RTN          ;21
06231 6024731777740000                  IZQ    OR,7777,6000          RTN          ;22
06232 2024731777700000                  IZQ    OR,7777,4000          RTN          ;23
06233 6024731777600000                  IZQ    OR,7777,0000          RTN          ;24
06234 2024731777400000                  IZQ    OR,7776,0000          RTN          ;25
06235 6024731777000000                  IZQ    OR,7774,0000          RTN          ;26
06236 2024731776000000                  IZQ    OR,7770,0000          RTN          ;27
06237 6024731774000000                  IZQ    OR,7760,0000          RTN          ;30
06240 2024731770000000                  IZQ    OR,7740,0000          RTN          ;31
06241 6024731760000000                  IZQ    OR,7700,0000          RTN          ;32
06242 2024731740000000                  IZQ    OR,7600,0000          RTN          ;33
06243 6024731700000000                  IZQ    OR,7400,0000          RTN          ;34
06244 2024731600000000                  IZQ    OR,7000,0000          RTN          ;35
06245 6024731400000000                  IZQ    OR,6000,0000          RTN          ;36
06246 2024731000000000                  IZQ    OR,4000,0000          RTN          ;37
06247 6024730000000000                  IZQ    OR,0000,0000          RTN          ;40
```

```
10137 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;  TEST OF ARITHMETIC AND LOGIC OPERATIONS
03
04                          B100:                                                    ;
05 C6250 4013730032002000           CZB   OR,.1B17,WRK0          POP                ; WRK0:= 8'100
06 C6251 4213551224003305 B101:     CABR  CAND,.1B1,STAT,STAT,CPUST PUSH            ; REPEAT
07                                                                                   ;   CLEAR ERROR FLAG , PUSH
08 C6252 4624730000143460           IZQ   OR,0,ARLOBASE/3-100                        ;   Q:= ADDR(RLOBASE) -100
09 C6253 4631001503000000           CXAQ  ADD,WRK0,F,F                               ;   EXECUTE TEST(Q+WRK0)
10 C6254 0605023600000000           AQ    SUB0,WRK2                                  ;   IF Q-WRK2<>0 THEN
11 C6255 4631000002426261           CALL  T,NZ,B100E                                 ;   BEGIN
12 C6256 4631000002006003           CALL  F,F,LOOP                                   ;     WRK1:=Q
13 C6257 4053503532003700 B102:     CAB   ADD0,.1,WRK0,WRK0      POP                 ;     TERROR
14 C6260 0631000000006251           JMP   F,F,B101                                   ;   END
15                                                                                   ;   TEST LOOPING
16 C6261 4607230033000000 B100E:    ZQB   OR,WRK1                                    ;   WRK0:= WRK0+2
17 C6262 4631000000006002           JMP   F,F,TERROR                                 ; UNTIL FALSE
18
19                          ARLOBASE:                                                ;
20 C6263 0607340034000000           ZBB   AND,WRK2                                   ;100  EXPECTED:= 0
21 C6264 2064700000000000           IZQ   ADD,0,0                RTN                 ;
22 C6265 2064722000000000           IZQ   SUB0,0,0               RTN                 ;102
23 C6266 0605230000000000           ZQ    OR                                         ;
24 C6267 0624731777777760           IZQ   OR,7777,7777                               ;104
25 C6270 6064623777777760           IQQ   SUB0,7777,7777         RTN                 ;
26 C6271 0653712034003700           CZB   SUN0,.1,WRK2                               ;106  EXPECTED:= -1
27 C6272 2064701777777760           IZQ   ADD,7777,7777          RTN                 ;
28 C6273 0624731777777760           IZQ   OR,7777,7777                               ;110
29 C6274 6064600000000000           IQQ   ADD,0,0                RTN                 ;
30 C6275 2064723777777760           IZQ   SUB0,7777,7777         RTN                 ;112
31 C6276 0605230000000000           ZQ    OR                                         ;
32 C6277 0624730525252520           IZQ   OR,2525,2525                               ;
33 C6300 6007230034000000           ZQB   OR,WRK2                RTN                 ;     EXPECTED:= 8'25252525
34 C6301 0624730525252520           IZQ   OR,2525,2525                               ;116
35 C6302 6064623252525240           IQQ   SUB0,5252,5252         RTN                 ;
36 C6303 0624731252525240           IZQ   OR,5252,5252                               ;
37 C6304 6007230034000000           ZQB   OR,WRK2                RTN                 ;     EXPECTED:= 8'52525252
```

```
10138 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 C6305 0624730525252520        IZQ   OR,2525,2525                    ;122
03 C6306 6064600525252520        IQQ   ADD,2525,2525          RTN       ;
04 C6307 0624730003770000        IZQ   OR,0017,7400                     ;      EXPECTED:= 8'177400
05 C6310 6007230034CC0000        ZQB   OR,WRK2                RTN       ;
06 C6311 0624730004C00020        IZQ   OR,0020,1                        ;126
07 C6312 2064601777767760        IQQ   ADD,7777,7377          RTN       ;
08 C6313 4624730010010040        IZQ   OR,0040,0402                     ;130
09 C6314 2064601773757740        IQQ   ADD,7757,6776          RTN       ;
10 C6315 0624730020030100        IZQ   OR,100,1404                      ;132
11 C6316 2064601763737700        IQQ   ADD,7717,5774          RTN       ;
12 C6317 4624730040070200        IZQ   OR,200,3410                      ;134
13 C6320 2064601743677600        IQQ   ADD,7617,3770          RTN       ;
14 C6321 0624730100170400        IZQ   OR,400,7420                      ;136
15 C6322 2064601703577400        IQQ   ADD,7416,7760          RTN       ;
16 C6323 4624730200371000        IZQ   OR,1001,7440                     ;140
17 C6324 2064601603377000        IQQ   ADD,7015,7740          RTN       ;
18 C6325 0624730400772000        IZQ   OR,2003,7500                     ;142
19 C6326 2064601402776000        IQQ   ADD,6013,7700          RTN       ;
20 C6327 4624731001774000        IZQ   OR,4007,7600                     ;144
21 C6330 2064601001774000        IQQ   ADD,4007,7600          RTN       ;
22 C6331 0624731774CC7760        IZQ   OR,7760,0377                     ;      EXPECTED:= 8'77600377
23 C6332 6007230034C00000        ZQB   OR,WRK2                RTN       ;
24 C6333 0624730004CC0020        IZQ   OR,20,0001                       ;150
25 C6334 2064622000C10000        IQQ   SUB0,0,400             RTN       ;
26 C6335 4624730010010040        IZQ   OR,40,0402                       ;152
27 C6336 2064622004C20020        IQQ   SUB0,20,1001           RTN       ;
28 C6337 0624730020030100        IZQ   OR,100,1404                      ;154
29 C6340 2064622014C40060        IQQ   SUB0,60,2003           RTN       ;
30 C6341 4624730040070200        IZQ   OR,200,3410                      ;156
31 C6342 2064622034100160        IQQ   SUB0,160,4007          RTN       ;
32 C6343 0624730100170400        IZQ   OR,400,7420                      ;160
33 C6344 2064622074200360        IQQ   SUB0,0361,17           RTN       ;
34 C6345 4624730200371000        IZQ   OR,1001,7440                     ;162
35 C6346 2064622174400760        IQQ   SUB0,0762,37           RTN       ;
36 C6347 0624730400772000        IZQ   OR,2003,7500                     ;164
37 C6350 2064622375CC1760        IQQ   SUB0,1764,77           RTN       ;
```

```
10139 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 C6351 4624731001774000        IZQ   OR,4007,7600                    ;166
03 C6352 2064622776CC3760        IQQ   SUB0,3770,0177        RTN         ;
04 C6353 0624730525252520        IZQ   OR,2525,2525                    ;     EXPECTED:= 8'25252525
05 C6354 6007230034CC0000        ZQB   OR,WRK2               RTN         ;
06 C6355 0624730525252520        IZQ   OR,2525,2525                    ;172
07 C6356 2024630525252520        IQQ   OR,2525,2525          RTN         ;
08 C6357 0624730525252520        IZQ   OR,2525,2525                    ;174
09 C6360 6024640525252520        IQQ   AND,2525,2525         RTN         ;
10 C6361 0624730525252520        IZQ   OR,2525,2525                    ;176
11 C6362 2024651252525240        IQQ   CAND,5252,5252        RTN         ;
12 C6363 0624731777777760        IZQ   OR,7777,7777                    ;     EXPECTED:= -1
13 C6364 6007230034CC0000        ZQB   OR,WRK2               RTN         ;
14 C6365 0624730525252520        IZQ   OR,2525,2525                    ;202
15 C6366 2024631252525240        IQQ   OR,5252,5252          RTN         ;
16 C6367 0624731252525240        IZQ   OR,5252,5252                    ;204
17 C6370 2024630525252520        IQQ   OR,2525,2525          RTN         ;
18 C6371 0624730525252520        IZQ   OR,2525,2525                    ;206
19 C6372 2024661252525240        IQQ   EXOR,5252,5252        RTN         ;
20 C6373 0624731252525240        IZQ   OR,5252,5252                    ;210
21 C6374 2024660525252520        IQQ   EXOR,2525,2525        RTN         ;
22 C6375 0624730525252520        IZQ   OR,2525,2525                    ;212
23 C6376 6024670525252520        IQQ   EXNOR,2525,2525       RTN         ;
24 C6377 0624731252525240        IZQ   OR,5252,5252                    ;214
25 C6400 6024671252525240        IQQ   EXNOR,5252,5252       RTN         ;
26 C6401 0624730000000000        IZQ   OR,0,0                          ;     EXPECTED:= 0
27 C6402 6007230034CC0000        ZQB   OR,WRK2               RTN         ;
28 C6403 0624730525252520        IZQ   OR,2525,2525                    ;220
29 C6404 6024641252525240        IQQ   AND,5252,5252         RTN         ;
30 C6405 0624731252525240        IZQ   OR,5252,5252                    ;222
31 C6406 6024640525252520        IQQ   AND,2525,2525         RTN         ;
32 C6407 0624730525252520        IZQ   OR,2525,2525                    ;224
33 C6410 2024650525252520        IQQ   CAND,2525,2525        RTN         ;
34 C6411 0624731252525240        IZQ   OR,5252,5252                    ;226
35 C6412 2024651252525240        IQQ   CAND,5252,5252        RTN         ;
36 C6413 0624730525252520        IZQ   OR,2525,2525                    ;230
37 C6414 2024660525252520        IQQ   EXOR,2525,2525        RTN         ;
```

```
10140 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 C6415 0624731252525240          IZQ   OR,5252,5252                        ;232
03 C6416 2024661252525240          IQQ   EXOR,5252,5252          RTN         ;
04 C6417 0624730525252520          IZQ   OR,2525,2525                        ;234
05 C6420 6024671252525240          IQQ   EXNOR,5252,5252         RTN         ;
06 C6421 0624731252525240          IZQ   OR,5252,5252                        ;236
07 C6422 6024670525252520          IQQ   EXNOR,2525,2525         RTN         ;
08 C6423 0624731252525240          IZQ   OR,5252,5252                        ;      EXPECTED:= 8'52525252
09 C6424 6007230034000000          ZQB   OR,WRK2                 RTN         ;
10 C6425 0624731252525240          IZQ   OR,5252,5252                        ;242
11 C6426 2024631252525240          IQQ   OR,5252,5252            RTN         ;
12 C6427 0624731252525240          IZQ   OR,5252,5252                        ;244
13 C6430 6024641252525240          IQQ   AND,5252,5252           RTN         ;
14 C6431 0624731252525240          IZQ   OR,5252,5252                        ;246
15 C6432 2024650525252520          IQQ   CAND,2525,2525          RTN         ;
16
17                         ;ALU REGISTER TEST
18
19 C6433 0024730000006000          IZQ   OR,0,300                POP         ;
20 C6434 0607230032000000          ZQB   OR,WRK0                             ; WRK0:= 8'300
21 C6435 0611730000001501          CZR   OR,.4,LAST                          ; LAST:= 4
22                         B300:                                             ; REPEAT
23 C6436 4613730034003700          CZB   OR,.1,WRK2                          ;   WRK2:= 1
24                         B310:                                             ;   REPEAT
25 C6437 4213551224003305          CABR  CAND,.1B1,STAT,STAT,CPUST PUSH      ;     CLEAR ERROR FLAG ,  PUSH
26 C6440 0624730000143240          IZQ   OR,0,REGBASE/3-300                  ;     Q:= ADDR(REGBASE)-300
27 C6441 4631001503000000          CXAQ  ADD,WRK0,F,F                        ;     EXECUTE TEST(Q+WRK0)
28 C6442 4607230033000000          ZQB   OR,WRK1                             ;     WRK1:= Q
29 C6443 4605123574000000          AB    SUB0,WRK1,WRK2                      ;     IF WRK1<>WRK2 THEN
30 C6444 0631000002426002          CALL  T,NZ,TERROR                         ;       TERROR
31 C6445 4631000002006003          CALL  F,F,LOOP                            ;     TEST LOOPING
32 C6446 4007101634000000 B315:    ABB   ADD,WRK2,WRK2           POP         ;     WRK2:= WRK2 SHIFT 1
33 C6447 0631000000426437          JMP   T,NZ,B310                           ;   UNTIL WRK2=0
34 C6450 4653503532003700 B320:    CAB   ADD0,.1,WRK0,WRK0                   ;   WRK0:= WRK0+2
35 C6451 4631000000006436          JMP   F,F,B300                            ; UNTIL FALSE
```

```
10141 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          REGBASE:                                        ;
03 C6452 6004431600C00000           ZAQ   OR,WRK2              RTN        ;300
04 C6453 0605230000C00000           ZQ    OR                              ;
05 C6454 0607431603C00000           ZAB   OR,WRK2,LAX                     ;302
06 C6455 2004431100C00000           ZAQ   OR,W2                RTN        ;
07 C6456 0607431603C00000           ZAB   OR,WRK2,LAX                     ;304
08 C6457 6004330003C00000           ZBQ   OR,LAX               RTN        ;
09 C6460 4607431620C00000           ZAB   OR,WRK2,W0                      ;306
10 C6461 6004431000C00000           ZAQ   OR,W0                RTN        ;
11 C6462 4607431620C00000           ZAB   OR,WRK2,W0                      ;310
12 C6463 2004330020C00000           ZBQ   OR,W0                RTN        ;
13 C6464 0607431621C00000           ZAB   OR,WRK2,W1                      ;312
14 C6465 2004431040C00000           ZAQ   OR,W1                RTN        ;
15 C6466 0607431621C00000           ZAB   OR,WRK2,W1                      ;314
16 C6467 6004330021C00000           ZBQ   OR,W1                RTN        ;
17 C6470 0607431622C00000           ZAB   OR,WRK2,W2                      ;316
18 C6471 2004431100C00000           ZAQ   OR,W2                RTN        ;
19 C6472 0607431622C00000           ZAB   OR,WRK2,W2                      ;320
20 C6473 6004330022C00000           ZBQ   OR,W2                RTN        ;
21 C6474 4607431623C00000           ZAB   OR,WRK2,W3                      ;322
22 C6475 6004431140C00000           ZAQ   OR,W3                RTN        ;
23 C6476 4607431623C00000           ZAB   OR,WRK2,W3                      ;324
24 C6477 2004330023C00000           ZBQ   OR,W3                RTN        ;
25 C6500 4607431625C00000           ZAB   OR,WRK2,CAUSE                   ;326
26 C6501 6004431240C00000           ZAQ   OR,CAUSE             RTN        ;
27 C6502 4607431625C00000           ZAB   OR,WRK2,CAUSE                   ;330
28 C6503 2004330025C00000           ZBQ   OR,CAUSE             RTN        ;
29 C6504 4607431626C00000           ZAB   OR,WRK2,SB                      ;332
30 C6505 6004431300C00000           ZAQ   OR,SB                RTN        ;
31 C6506 4607431626C00000           ZAB   OR,WRK2,SB                      ;334
32 C6507 2004330026C00000           ZBQ   OR,SB                RTN        ;
33 C6510 0607431627C00000           ZAB   OR,WRK2,INF                     ;336
34 C6511 2004431340C00000           ZAQ   OR,INF               RTN        ;
35 C6512 0607431627C00000           ZAB   OR,WRK2,INF                     ;340
36 C6513 6004330027C00000           ZBQ   OR,INF               RTN        ;
37 C6514 0607431630C00000           ZAB   OR,WRK2,.2                      ;342
38 C6515 2004431400C00000           ZAQ   OR,.2                RTN        ;
```

```
10142 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 C6516 0607431630CC0000        ZAB   OR,WRK2,.2                    ;344
03 C6517 6004330030CC0000        ZBQ   OR,.2                 RTN     ;
04 C6520 0607431633CC0000        ZAB   OR,WRK2,WRK1                  ;346
05 C6521 2004431540CC0000        ZAQ   OR,WRK1               RTN     ;
06 C6522 0607431633CC0000        ZAB   OR,WRK2,WRK1                  ;350
07 C6523 6004330033CC0000        ZBQ   OR,WRK1               RTN     ;
08 C6524 4607431634CC0000        ZAB   OR,WRK2,WRK2                  ;352
09 C6525 6004431600CC0000        ZAQ   OR,WRK2               RTN     ;
10 C6526 4607431634CC0000        ZAB   OR,WRK2,WRK2                  ;354
11 C6527 2004330034CC0000        ZBQ   OR,WRK2               RTN     ;
12 C6530 0607431635CC0000        ZAB   OR,WRK2,WRK3                  ;356
13 C6531 2004431640CC0000        ZAQ   OR,WRK3               RTN     ;
14 C6532 0607431635CC0000        ZAB   OR,WRK2,WRK3                  ;360
15 C6533 6004330035CC0000        ZBQ   OR,WRK3               RTN     ;
16 C6534 0607431636CC0000        ZAB   OR,WRK2,WRK4                  ;362
17 C6535 2004431700CC0000        ZAQ   OR,WRK4               RTN     ;
18 C6536 0607431636CC0000        ZAB   OR,WRK2,WRK4                  ;364
19 C6537 6004330036CC0000        ZBQ   OR,WRK4               RTN     ;
20 C6540 4607431637CC0000        ZAB   OR,WRK2,WRK5                  ;366
21 C6541 6004431740CC0000        ZAQ   OR,WRK5               RTN     ;
22 C6542 4607431637CC0000        ZAB   OR,WRK2,WRK5                  ;370
23 C6543 2004330037CC0000        ZBQ   OR,WRK5               RTN     ;
24 C6544 0605431600CC0004        ZAR   OR,WRK2,LCD                   ;372
25 C6545 2004730000003000        RZQ   OR,LCS                RTN     ;
26 C6546 4003730037CC1700        SZB   OR,17,WRK5            POP     ; SAVE RUN COUNTER IN WRK5
27 C6547 4631000000CC6450        JMP   F,F,B320                      ; RETURN
28 C6550 4641421600CC0002        ZAS   SUB,WRK2,0                    ;376
29 C6551 6000730000CC0000        SZQ   OR,0                  RTN     ;
30 C6552 0641421600CC0102        ZAS   SUB,WRK2,1                    ;400
31 C6553 2000730000000100        SZQ   OR,1                  RTN     ;
32 C6554 0641421600CC0202        ZAS   SUB,WRK2,2                    ;402
33 C6555 2000730000CC0200        SZQ   OR,2                  RTN     ;
34 C6556 4641421600CC0302        ZAS   SUB,WRK2,3                    ;404
35 C6557 6000730000CC0300        SZQ   OR,3                  RTN     ;
36 C6560 0641421600CC0402        ZAS   SUB,WRK2,4                    ;406
37 C6561 2000730000CC0400        SZQ   OR,4                  RTN     ;
```

```
10143 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 C6562 4641421600CC0502        ZAS   SUB,WRK2,5                        ;410
03 C6563 6000730000CC0500        SZQ   OR,5                      RTN     ;
04 C6564 4641421600CC0602        ZAS   SUB,WRK2,6                        ;412
05 C6565 6000730000CC0600        SZQ   OR,6                      RTN     ;
06 C6566 0641421600CC0702        ZAS   SUB,WRK2,7                        ;414
07 C6567 2000730000CC0700        SZQ   OR,7                      RTN     ;
08 C6570 0641421600CC1002        ZAS   SUB,WRK2,10                       ;416
09 C6571 2000730000CC1000        SZQ   OR,10                     RTN     ;
10 C6572 4641421600CC1102        ZAS   SUB,WRK2,11                       ;420
11 C6573 6000730000CC1100        SZQ   OR,11                     RTN     ;
12 C6574 4641421600CC1202        ZAS   SUB,WRK2,12                       ;422
13 C6575 6000730000CC1200        SZQ   OR,12                     RTN     ;
14 C6576 0641421600CC1302        ZAS   SUB,WRK2,13                       ;424
15 C6577 2000730000CC1300        SZQ   OR,13                     RTN     ;
16 C6600 4641421600CC1402        ZAS   SUB,WRK2,14                       ;426
17 C6601 6000730000CC1400        SZQ   OR,14                     RTN     ;
18 C6602 0641421600CC1502        ZAS   SUB,WRK2,15                       ;430
19 C6603 2000730000CC1500        SZQ   OR,15                     RTN     ;
20 C6604 0641421600CC1602        ZAS   SUB,WRK2,16                       ;432
21 C6605 2000730000CC1600        SZQ   OR,16                     RTN     ;
22 C6606 4641421600CC1702        ZAS   SUB,WRK2,17                       ;434
23 C6607 6000730000CC1700        SZQ   OR,17                     RTN     ;
24                                                                       ; TEST OF IC MUST BE LAST
25 C6610 4053503532CC3700        CAB   ADDO,.1,WRK0,WRK0         POP     ;  SKIP PATTERN=1
26 C6611 0631000000CC6446        JMP   F,F,B315                          ;
27 C6612 4605431600CC0003        ZAR   OR,WRK2,ICD                       ; 440
28 C6613 0604730000CC1000        RZQ   OR,ICS                            ;
29 C6614 4611541600CC3500        CA    AND,.1B0,WRK2                     ; STOP WHEN WRK2= 8'40000000
30 C6615 2031000000426616        JMP   T,NZ,.+3                  RTN     ;
31 C6616 4041421740CC1702        ZAS   SUB,WRK5,17               POP     ; RESTORE RUN COUNTER
32 C6617 0653702030CC3700        CZB   ADDO,.1,.2                        ; .2:= 2
```

```
10144 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;TEST OF SHIFT FUNCTIONS
03
04                          B500:                                                ;B500:
05 C6620 0624730000012000           1ZQ    OR,0,500                            ; WRK0:= 8'500
06 C6621 0607230032000000           ZQB    OR,WRK0                             ;
07 C6622 4213551224003305 B501:     CABR   CAND,.1B1,STAT,STAT,CPUST PUSH       ; REPEAT
08                                                                              ;   CLEAR ERROR FLAG,  PUSH
09 C6623 0624730000143560           1ZQ    OR,0,SHIFTBASE/3-500                ;   Q:= ADDR(SHIFTBASE)-500
10 C6624 4631001503000000           CXAQ   ADD,WRK0,F,F                        ;   EXECUTE TEST(Q+WRK0)
11 C6625 4605123574000000           AB     SUB0,WRK1,WRK2                      ;   IF WRK1<>WRK2 THEN
12 C6626 0631000002426002           CALL   T,NZ,TERROR                         ;     TERROR
13 C6627 4631000002C06003           CALL   F,F,LOOP                            ;   TEST LOOPING
14 C6630 0007302032000000           ZBB    ADD0,WRK0                 POP       ;   WRK0:= WRK0+1
15 C6631 0631000000006622           JMP    F,F,B501                            ; UNTIL FALSE
16
17                          TB500:                                               ;500: TEST DOUBLE SHIFT
18 C6632 4624731000C00000           1ZQ    OR,4000,0                           ; Q:= 8'40000000
19 C6633 4607340033C00000           ZBB    AND,WRK1                            ; WRK1:= 0
20 C6634 4613730034003700           CZB    OR,.1,WRK2                          ; WRK2:= 1  C. EXPECTED DATA
21 C6635 2036330033C00060           DLZBB  OR,WRK1,Z,NL,F,F          RTN        ; WRK1,Q:= (WRK1,Q) SHIFT 1
22                                                                              ; RETURN
23
24                          TB501:                                               ;501: SI=ZERO LEFT SHIFT
25 C6636 4653712033003700           CZB    SUN0,.1,WRK1                        ; WRK1:= -1
26 C6637 0613551574003700           CAB    CAND,.1,WRK1,WRK2                   ; WRK2:= 8'77777776  C. EXPECTED DATA
27 C6640 6037330033C00060           SLZBB  OR,WRK1,Z,NL,F,F          RTN        ; WRK1:= WRK1(1:23) CON 1 EXT 0
28                                                                              ; RETURN
29
30                          TB502:                                               ;502: SI=ZERO  RIGHT SHIFT
31 C6641 4653712033003700           CZB    SUN0,.1,WRK1                        ; WRK1:= -1
32 C6642 4613551574003500           CAB    CAND,.1B0,WRK1,WRK2                 ; WRK2:= 1 EXT 0 CON WRK1(1:23)
33                                                                              ; C.  EXPECTED DATA
34 C6643 2035330033000060           SRZBB  OR,WRK1,Z,NL,F,F          RTN        ; WRK1:= 1 EXT 0 WRK1(1:22)
35                                                                              ;RETURN
```

```
10145 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          TB503:                                          ;503: SI=LINK  LEFT SHIFT
03 C6644 4613730034CC3700          CZB   OR,.1,WRK2                         ; WRK2:= 1  C. EXPECTED DATA
04 C6645 4613730033CC3500          CZB   OR,.1B0,WRK1                       ; WRK1:= 1B0
05 C6646 0637330033CC0060          SLZBB OR,WRK1,Z,NL,F,F                   ; WRK1:= WRK1(1:23) CON 1 EXT 0
06 C6647 2037330033CC1060          SLZBB OR,WRK1,LNK,NL,F,F      RTN        ; WRK1:= WRK1(1:23) CON 1 EXT LINK
07                                                                          ; RETURN
08
09                          TB504:                                          ;504: SI=LINK  RIGHT SHIFT
10 C6650 0613730034CC3500          CZB   OR,.1B0,WRK2                       ; WRK2:= 1B0
11 C6651 0613730033CC3700          CZB   OR,.1,WRK1                         ; WRK1:= 1
12 C6652 4635330033CC0060          SRZBB OR,WRK1,Z,NL,F,F                   ; WRK1:= 1 EXT 0 CON WRK1(0:22)
13 C6653 6035330033CC1060          SRZBB OR,WRK1,LNK,NL,F,F      RTN        ; WRK1:= 1 EXT LINK CON WRK1(0:22)
14                                                                          ; RETURN
15
16                          TB505:                                          ;505: SI=ADC  LEFT SHIFT
17 C6654 4613730034CC3700          CZB   OR,.1,WRK2                         ; WRK2:= 1
18 C6655 0604240000CC0000          ZQQ   AND                                ; Q:= 0
19 C6656 4674123573CC0040          DRABB SUB0,WRK1,WRK1,Z,MC,F,F            ; WRK1:= 0, ADC:= -,Q(23)
20 C6657 2037330033CC2060          SLZBB OR,WRK1,ADC,NL,F,F      RTN        ; WRK1:= WRK1(1:23) CON 1 EXT ADC
21                                                                          ; RETURN
22
23                          TB506:                                          ;506: SI=F(0)  RIGHT SHIFT
24 C6660 0624731400CC0000          IZQ   OR,6000,0                          ;
25 C6661 0607230034CC0000          ZQB   OR,WRK2                            ; WRK2:= 8'60000000
26 C6662 4613730033CC3500          CZB   OR,.1B0,WRK1                       ; WRK1:= 1B0
27 C6663 6035330033CC2060          SRZBB OR,WRK1,F0,NL,F,F       RTN        ; WRK1:= 1 EXT WRK1(0) CON WRK1(0:22)
28                                                                          ; RETURN
29
30                          TB507:                                          ;507: SI=F(0) EXOR OVERFLOW  RIGHT SHIFT
31 C6664 4613730033CC3500          CZB   OR,.1B0,WRK1                       ; WRK1:= WRK2:= 1B0
32 C6665 0607431574CC0000          ZAB   OR,WRK1,WRK2                       ;
33                                                                          ; WRK1:= WRK1+WRK2
34 C6666 6075101633CC3060          SRABB ADD,WRK2,WRK1,SGN,NL,F,F RTN       ; WRK1:= 1 EXT(F0 EXOR OVERFLOW)
35                                                                          ;         CON WRK1(0:22)
36                                                                          ; RETURN
```

```

                               SHIFTBASE:                                                    ;
   C6667 463100000CC6632              JMP    F,F,TB500                                       ;
   C6670 063100000CC6636              JMP    F,F,TB501                                       ;
   C6671 063100000CC6641              JMP    F,F,TB502                                       ;
   C6672 063100000CC6644              JMP    F,F,TB503                                       ;
   C6673 063100000CC6650              JMP    F,F,TB504                                       ;
   C6674 463100000CC6654              JMP    F,F,TB505                                       ;
   C6675 063100000CC6660              JMP    F,F,TB506                                       ;
   C6676 463100000CC6664              JMP    F,F,TB507                                       ;

                               ;TEST OF JUMP CONDITIONS
                               B540:                                                         ;
   C6677 402473000C013000             IZQ    OR,0,540                POP                     ; WRK0:= 8'540
   C6700 0607230032CC0000             ZQB    OR,WRK0                                         ;
                               B541:                                                         ; REPEAT
   C6701 4213551224CC3305             CABR   CAND,.1B1,STAT,STAT,CPUST PUSH                  ;   CLEAR ERROR FLAG  ,PUSH
   C6702 062473000C145420             IZQ    OR,0,JUMPBASE/3-540                             ;   Q:= ADDR(JUMPBASE)-540
   C6703 46310015C3CC0000             CXAQ   ADD,WRK0,F,F                                    ;   EXECUTE TEST(Q+WRK0)
   C6704 4631000002C06003             CALL   F,F,LOOP                                        ;   TEST LOOPING
   C6705 0007302032C00000             ZBB    ADD0,WRK0               POP                     ;   WRK0:= WRK0+1
   C6706 063100000CC06701             JMP    F,F,B541                                        ; UNTIL FALSE

                               TNEG:                                                         ;540: DBUS(0)
   C6707 0607340034C00000             ZBB    AND,WRK2                                        ; WRK2:= 0
   C6710 4631000000416002             JMP    T,NEG,TERROR                                    ; IF WRK2<0 THEN GOTO TERROR
   C6711 06137300340C3500             CZB    OR,.1B0,WRK2                                    ; IF -,8'40000000<0 THEN
   C6712 6031000000016002             JMP    F,NEG,TERROR            RTN                     ;   GOTO TERROR
                                                                                             ; RETURN

                               TNZ:                                                          ;541: F<>0
   C6713 0607340034CC0000             ZBB    AND,WRK2                                        ; WRK2:= 0
   C6714 4631000000426002             JMP    T,NZ,TERROR                                     ; IF WRK2<>0 THEN GOTO TERROR
   C6715 0607302034CC0000             ZEB    ADD0,WRK2                                       ; WRK2:= 1
   C6716 603100000C026002             JMP    F,NZ,TERROR             RTN                     ; IF -,WRK2<>0 THEN GOTO TERROR
                                                                                             ; RETURN
```

```
10147 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          TOVFL:                                          ;542: ARITHMETIC OVERFLOW
03 C6717 4653712034CC3500           CZB   SUNO,.1BU,WRK2                   ; WRK2:= -8'40000000
04 C6720 4631000000C36002           JMP   F,OVFL,TERROR                    ; IF -,OVERFLOW THEN GOTO TERROR
05 C6721 0653702034CC3500           CZB   ADDO,.1BU,WRK2                   ; WRK2:= 8'40000000 + 1
06 C6722 6031000000436002           JMP   T,OVFL,TERROR           RTN      ; IF OVERFLOW THEN GOTO TERROR
07                                                                         ; RETURN
08
09                          TCARRY:                                         ;543: CARRY
10 C6723 0624731777777760           IZQ   OR,7777,7777                     ; WRK2:= 8'77777777 + 1
11 C6724 4607202034CC0000           ZQB   ADDO,WRK2                        ;
12 C6725 0631000000046002           JMP   F,CARRY,TERROR                   ; IF -,CARRY THEN GOTO TERROR
13 C6726 0607302034CC0000           ZBB   ADDO,WRK2                        ; WRK2:= WRK2 + 1
14 C6727 2031000000446002           JMP   T,CARRY,TERROR          RTN      ; IF CARRY THEN GOTO TERROR
15                                                                         ; RETURN
16
17                          TSIGN:                                          ;544: F(0) EXOR OVERFLOW
18 C6730 0653720034CC3500           CZB   SUB,.1BU,WRK2                    ; WRK2:= 8'40000000 - 1
19 C6731 4631000000056002           JMP   F,SIGN,TERROR                    ; IF -,F(0) EXOR OVERFLOW THEN GOTO TERROR
20 C6732 0607302034CC0000           ZBB   ADDO,WRK2                        ; WRK2:= WRK2 + 1
21 C6733 6031000000456002           JMP   T,SIGN,TERROR           RTN      ; IF F(0) EXOR OVERFLOW THEN GOTO TERROR
22                                                                         ; RETURN
23
24                          TNORU:                                          ;545: DBUS(0)<>DBUS(1)
25 C6734 0624731400CC0000           IZQ   OR,6000,0                        ; WRK2:= 8'60000000
26 C6735 0607230034CC0000           ZQB   OR,WRK2                          ;
27 C6736 0631000000466002           JMP   T,NORM0,TERROR                   ; IF DBUS(0)<>DBUS(1) THEN GOTO TERROR
28 C6737 0607340034CC0000           ZBB   AND,WRK2                         ; WRK2:= 0
29 C6740 0631000000466002           JMP   T,NORM0,TERROR                   ; IF DBUS(0)<>DBUS(1) THEN GOTO TERROR
30 C6741 0613730034CC3500           CZB   OR,.1B0,WRK2                     ; WRK2:= 8'40000000
31 C6742 4631000000C66002           JMP   F,NORM0,TERROR                   ; IF -,DBUS(0)<>DBUS(1) THEN GOTO TERROR
32 C6743 0613730034CC3300           CZB   OR,.1B1,WRK2                     ; WRK2:= 8'20000000
33 C6744 2031000000C66002           JMP   F,NORM0,TERROR          RTN      ; IF -,DBUS(0)<>DBUS(1) THEN GOTO TERROR
34                                                                         ; RETURN
```

```
10148 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          TNOR1:                                          ;546: DBUS(1)<>DBUS(2)
03 C6745 0624730600C00000           IZQ    OR,3000,0                        ; WRK2:= 8'30000000
04 C6746 0607230034C00000           ZQB    OR,WRK2                          ;
05 C6747 4631000000476002           JMP    T,NORM1,TERROR                   ; IF DBUS(1)<>DBUS(2) THEN GOTO TERROR
06 C6750 0607340034C00000           ZBB    AND,WRK2                         ; WRK2:= 0
07 C6751 4631000000476002           JMP    T,NORM1,TERROR                   ; IF DBUS(1)<>DBUS(2) THEN GOTO TERROR
08 C6752 0613730034CC3300           CZB    OR,.1B1,WRK2                     ; WRK2:= 8'20000000
09 C6753 0631000000076002           JMP    F,NORM1,TERROR                   ; IF -<DBUS(1)<>DBUS(2) THEN GOTO TERROR
10 C6754 4613730034CC3200           CZB    OR,.1B2,WRK2                     ; WRK2:= 8'10000000
11 C6755 6031000000076002           JMP    F,NORM1,TERROR          RTN      ; IF -, DBUS(1)<>DBUS(2) THEN GOTO TERROR
12                                                                          ; RETURN
13
14                          TDIVOF:                                         ;547: LINK<>CARRY
15 C6756 0613730034CC3500           CZB    OR,.1B0,WRK2                     ; WRK2:= 8'40000000
16 C6757 4677302034CC0060           SLZBB  ADD0,WRK2,Z,NL,F,F               ; WRK2:= (WRK2+1) SHIFT 1
17 C6760 0631000000106002           JMP    F,DIVOF,TERROR                   ; IF -,LINK<>CARRY THEN GOTO TERROR
18 C6761 0653712034CC3700           CZB    SUN0,.1,WRK2                     ; WRK2:= -1
19 C6762 0677101634CC006U           SLABB  ADD,WRK2,WRK2,Z,NL,F,F           ; WRK2:= (WRK2+WRK2) SHIFT 1
20 C6763 2031000000506002           JMP    T,DIVOF,TERROR          RTN      ; IF LINK<>CARRY THEN GOTO TERROR
21                                                                          ; RETURN
22
23                          TLCN:                                           ;550:  LC(0)
24 C6764 4607340034CC0004           ZBBR   AND,WRK2,LCD                     ; WRK2:=LCD:= 0
25 C6765 0631000000516002           JMP    T,LCN,TERROR                     ; IF LC(0) THEN GOTO TERROR
26 C6766 4613730034CC3504           CZBR   OR,.1B0,WRK2,LCD                 ; WRK2:=LCD:= 8'40000000
27 C6767 2031000000116002           JMP    F,LCN,TERROR            RTN      ; IF -,LC(0) THEN GOTO TERROR
28                                                                          ; RETURN
29
30                          TMAXL:                                          ;551: LC(0)=0 OR MAXLOOP(0)=0
31 C6770 0624730000001400           IZQ    OR,0,60                          ; Q:= 48
32 C6771 0607230034C00000           ZQB    OR,WRK2                          ; WRK2:= 48
33 C6772 0211730000CC3504           CZR    OR,.1B0,LCD             PUSH     ; LCD:= 8'40000000 , PUSH
34 C6773 0631000000526777           JMP    T,MAXLO,TMAXE                    ; IF LC(0)=0 OR MAXLOOP(0)=0
35                                                                          ; THEN GOTO TMAXERROR
36 C6774 0607310034C00000           ZBB    SUN,WRK2                         ; WRK2:= WRK2-1
37 C6775 2637330034C24060           NSZBB  OR,WRK2,NL,F,NZ         LRTN     ; UNTIL WRK2=0
38 C6776 2031000000126002           JMP    F,MAXLO,TERROR          RTN      ; IF -,LC(0)=0 OR MAXLOOP(0)=0 THEN
39                                                                          ; GOTO TERROR
```

```
10149 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          TMAXE:                                          ; RETURN
03 C6777 0005230000CC0000           ZQ     OR                       POP     ;
04 C7000 4631000000C6002            JMP    F,F,TERROR                       ;
05
06                          TNWADR:                                         ;552: LOGADDR(0:20)<>0
07 C7001 4607340034CC0001           ZBBR   AND,WRK2,LAST                    ; WRK2:=LAST:= 0
08 C7002 4631000000536002           JMP    T,NWADR,TERROR                   ; IF LOGADR(0:20)<>0 THEN GOTO TERROR
09 C7003 4613730034CC3601           CZBR   OR,.1B20,WRK2,LAST               ; WRK2:=LAST:= .1B20
10 C7004 6031000000136002           JMP    F,NWADR,TERROR           RTN     ; IF -,LOGADR(0:20)<>0 THEN GOTO TERROR
11                                                                          ; RETURN
12
13                          TNGADR:                                         ;553:  LOGADDR(0)=1
14 C7005 4607340034CC0001           ZBBR   AND,WRK2,LAST                    ; WRK2:=LAST:= 0
15 C7006 0631000000546002           JMP    T,NGADR,TERROR                   ; IF LOGADDR(0)=1 THEN GOTO TERROR
16 C7007 4613730034CC3501           CZBR   OR,.1B0,WRK2,LAST                ; WRK2:=LAST:= 1B0
17 C7010 2031000000146002           JMP    F,NGADR,TERROR           RTN     ; IF -,LOGADDR(0)=1 THEN GOTO TERROR
18                                                                          ; RETURN
19
20                          TMMODE:                                         ;554:  MONITORMODE
21 C7011 0613730034CC3505           CZBR   OR,.1B0,WRK2,CPUST               ; WRK2:=CPUST:= .1B0
22 C7012 0631000000166002           JMP    F,MMODE,TERROR                   ; IF -,MONITORMODETHEN GOTO TERROR
23 C7013 0607340034CC0005           ZBBR   AND,WRK2,CPUST                   ; WRK2:=CPUST:= 0
24 C7014 2031000000566002           JMP    T,MMODE,TERROR           RTN     ; IF MONITORMODE THEN GOTO TERROR
25                                                                          ; RETURN
26
27                          TEMODE:                                         ;555:  ESCAPEMODE
28 C7015 0613730034CC3305           CZBR   OR,.1B1,WRK2,CPUST               ; WRK2:=CPUST:= 1B1
29 C7016 4631000000176002           JMP    F,EMODE,TERROR                   ; IF -,ESCAPEMODE THEN GOTO TERROR
30 C7017 0607340034CC0005           ZBBR   AND,WRK2,CPUST                   ; WRK2:=CPUST:= 0
31 C7020 6031000000576002           JMP    T,EMODE,TERROR           RTN     ; IF ESCAPEMODE THEN GOTO TERROR
32                                                                          ; RETURN
```

```
10150 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                              JUMPBASE:                                       ;
03 C7021 0631000000006707              JMP    F,F,TNEG                         ;
04 C7022 0631000000006713              JMP    F,F,TNZ                          ;
05 C7023 4631000000006717              JMP    F,F,TOVFL                        ;
06 C7024 0631000000006723              JMP    F,F,TCARRY                       ;
07 C7025 4631000000006730              JMP    F,F,TSIGN                        ;
08 C7026 0631000000006734              JMP    F,F,TNOR0                        ;
09 C7027 0631000000006745              JMP    F,F,TNOR1                        ;
10 C7030 4631000000006756              JMP    F,F,TDIVOF                       ;
11 C7031 0631000000006764              JMP    F,F,TLCN                         ;
12 C7032 0631000000006770              JMP    F,F,TMAXL                        ;
13 C7033 0631000000007001              JMP    F,F,TNWADR                       ;
14 C7034 4631000000007005              JMP    F,F,TNGADR                       ;
15 C7035 4631000000007011              JMP    F,F,TMMODE                       ;
16 C7036 0631000000007015              JMP    F,F,TEMODE                       ;
```

```
10151 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;TEST OF HALFWORD MANIPULATOR
03
04 C7037 0024730000014000          IZQ   OR,0,600                 POP          ; WRK0:= 8'600
05 C7040 0607230032000000          ZQB   OR,WRK0                               ; FOR FUNC:=HLOAD,EXLOAD,HSTOR,MASK,
06 C7041 4624730000162520          IZQ   OR,0,HWBASE/3                         ;   EXT,SWAP,LSWAP,RSWAP DO
07 C7042 0607230037000000          ZQB   OR,WRK5                               ; BEGIN
08 C7043 0613730036003700 B601:    CZB   OR,.1,WRK4                            ;   FOR ODD:= 1,0 DO
09                        B602:                                                ;   BEGIN
10 C7044 4605431700000001          ZAR   OR,WRK4,LAST                          ;     FOR DATA:= 8'7777,8'77770000 DO
11 C7045 4613730021000300          CZB   OR,B12,W1                             ;     BEGIN
12 C7046 4213551224003305 B603:    CABR  CAND,.1B1,STAT,STAT,CPUST  PUSH       ;       SET UP LOOP
13 C7047 4624730000145520          IZQ   OR,0,HWRES/3-600                      ;       IF RESULT(FUNC)<>
14 C7050 0631431743000000          CXZA  OR,WRK5,F,F                           ;       EXPECTED(WRK0) THEN
15 C7051 4631001503000000          CXAQ  ADD,WRK0,F,F                          ;         TERROR
16 C7052 4605123574000000          AE    SUB0,WRK1,WRK2                        ;
17 C7053 0631000002426002          CALL  T,NZ,TERROR                           ;
18 C7054 4631000002006003          CALL  F,F,LOOP                              ;       TEST LOOPING
19 C7055 0007302032000000          ZBB   ADD0,WRK0                POP          ;       WRK0:= WRK0+1
20 C7056 4607320021000000          ZBB   SUB,W1                                ;     END FOR DATA
21 C7057 0651523040000300          CA    SUB0,B12,W1                           ;
22 C7060 0631000000427046          JMP   T,NZ,B603                             ;
23 C7061 4607310036000000          ZBB   SUN,WRK4                              ;   END FOR ODD
24 C7062 0631000000027044          JMP   F,NZ,B602                             ;
25 C7063 0607302037000000          ZBB   ADD0,WRK5                             ; END FOR FUNC
26 C7064 0631000000007043          JMP   F,F,B601                              ;
27
28                          HWRES:                                           ; TABLE OF EXPECTED DATA
29 C7065 6013730034000300          CZB   OR,B12,WRK2              RTN          ;600: HLOAD,ODD=1,00007777
30 C7066 6007340034000000          ZBB   AND,WRK2                 RTN          ;601:               77770000
31 C7067 6007340034000000          ZBB   AND,WRK2                 RTN          ;602:       ODD=0,00007777
32 C7070 6013730034000300          CZB   OR,B12,WRK2              RTN          ;603:               77770000
33 C7071 6053712034003700          CZB   SUN0,.1,WRK2             RTN          ;604: EXLOA,ODD=1,00007777
34 C7072 6007340034000000          ZBB   AND,WRK2                 RTN          ;605:               77770000
35 C7073 6007340034000000          ZBB   AND,WRK2                 RTN          ;606:       ODD=0,00007777
36 C7074 6053712034003700          CZB   SUN0,.1,WRK2             RTN          ;607:              77770000
37 C7075 6013730034000300          CZB   OR,B12,WRK2              RTN          ;610: HSTOR,ODD=1,00007777
```

10152 CPU823 MICROPROGRAM LBJ 850816 VERSION 8

```
C7076 6007340034C00000         ZBB   AND,WRK2            RTN        ;611:              77770000
C7077 6053710034C00300         CZB   SUN,B12,WRK2        RTN        ;612:        ODD=0,00007777
C7100 6007340034C00000         ZBB   AND,WRK2            RTN        ;613:              77770000
C7101 6053710034C00300         CZB   SUN,B12,WRK2        RTN        ;614: MASK  ,ODD=1,00007777
C7102 6053710034C00300         CZB   SUN,B12,WRK2        RTN        ;615:              77770000
C7103 6013730034C00300         CZB   OR,B12,WRK2         RTN        ;616:        ODD=0,00007777
C7104 6013730034C00300         CZB   OR,B12,WRK2         RTN        ;617:              77770000
C7105 6053712034C03700         CZB   SUNO,.1,WRK2        RTN        ;620: EXT   ,ODD=1,00007777
C7106 6007340034C00000         ZBB   AND,WRK2            RTN        ;621:              77770000
C7107 6053712034C03700         CZB   SUNO,.1,WRK2        RTN        ;622:        ODD=0,00007777
C7110 6007340034C00000         ZBB   AND,WRK2            RTN        ;623:              77770000
C7111 6053710034C00300         CZB   SUN,B12,WRK2        RTN        ;624: SWAP  ,ODD=1,00007777
C7112 6013730034C00300         CZB   OR,B12,WRK2         RTN        ;625:              77770000
C7113 6053710034C00300         CZB   SUN,B12,WRK2        RTN        ;626:        ODD=0,00007777
C7114 6013730034C00300         CZB   OR,B12,WRK2         RTN        ;627:              77770000
C7115 6007340034C00000         ZBB   AND,WRK2            RTN        ;630: LSWAP,ODD=1,00007777
C7116 6013730034C00300         CZB   OR,B12,WRK2         RTN        ;631:              77770000
C7117 6007340034C00000         ZBB   AND,WRK2            RTN        ;632:        ODD=0,00007777
C7120 6013730034C00300         CZB   OR,B12,WRK2         RTN        ;633:              77770000
C7121 6053710034C00300         CZB   SUN,B12,WRK2        RTN        ;634: RSWAP,ODD=1,00007777
C7122 6007340034C00000         ZBB   AND,WRK2            RTN        ;635:              77770000
C7123 6053710034C00300         CZB   SUN,B12,WRK2        RTN        ;636:        ODD=0,00007777
C7124 6007340034C00000         ZBB   AND,WRK2            RTN        ;637:              77770000

                       HWBASE:                                      ;
C7125 2062731073C00000         HZB   OR,HLOAD,W1,WRK1    RTN        ;
C7126 6062731073C00100         HZB   OR,EXLOAD,W1,WRK1   RTN        ;
C7127 6062731073C00200         HZB   OR,HSTOK,W1,WRK1    RTN        ;
C7130 2062731073C00300         HZB   OR,MASK,W1,WRK1     RTN        ;
C7131 6062731073C00400         HZB   OR,EXT,W1,WRK1      RTN        ;
C7132 2062731073C00500         HZB   OR,SWAP,W1,WRK1     RTN        ;
C7133 2062731073C00600         HZB   OR,LSWAP,W1,WRK1    RTN        ;
C7134 6062731073C00700         HZB   OR,RSWAP,W1,WRK1    RTN        ;
```

```
                                 ;TEST OF EXTERNAL INTERRUPTS

                                 B700:                                                  ;
   07135 402473000000100                IZQ   OR,0,4                  POP             ; IF CPU822 NOT AVAILABLE THEN
   07136 060564000005000                RQ    AND,SWTCH                               ;   GOTO CPULOOP
   07137 063100000426167                JMP   T,NZ,CPULP                              ;
   07140 463100002004016                CALL  F,F,CINIR                               ; CLEAR EXTERNAL INTERRUPTS
   07141 062473001761760                IZQ   OR,77,77                                ;
   07142 060523000000021                ZQR   OR,ILIM                                 ; ILIM:= 63,63 C. ALLOW ALL INTERRUPTS
   07143 461473000000100                XZQ   OR,CIOSTA                               ; Q:= CPUIOSTATUS
   07144 462464007600000                IQQ   AND,0370,0                              ; Q:= CPUIOSTATUS(4:8)
   07145 061363003702300                CQB   OR,.1B12,WRK5                           ; WRK5:= 1B12 OR Q
   07146 466273177700500                HZB   OR,SWAP,WRK5,WRK5                       ; WRK5:= WRK5(12:23) CON WRK5(0:11)
   07147 060712163400000                ABB   SUB,WRK2,WRK2                           ; FOR WRK2:= 0 STEP 1 UNTIL 7 DO
   07150 464740363400012 B750:          ZABR  ADDO,WRK2,WRK2,DATO                     ; BEGIN
                                                                                        ;   DATAOUT:= WRK2
   07151 421355122403305                CABR  CAND,.1B1,STAT,STAT,CPUST PUSH          ;   CLEAR ERRORFLAG  ,PUSH
   07152 060543174023001 1              ZAR   OR,WRK5,LAD             CONT WRT        ;   LAD:= WRK5  ,WRT
   07153 062473000016020                IZQ   OR,0,701                                ;
   07154 060723003200000                ZQB   OR,WRKO                                 ;   WRKO:= 8'701
   07155 065773003300100                XZB   OR,CIOST,WRK1           CONT 0 1        ;   WRK1:= CPUIOSTATUS
   07156 061354157303400                CAB   AND,.7,WRK1,WRK1                        ;   IF IOSTATUS(21:23) <> 0  THEN
   07157 063100002426163                CALL  T,NZ,TBERR                              ;     TBERR
   07160 063100002617241                CALL  T,EXINT,CHINT                           ;   IF EXINT THEN
                                                                                        ;     CHECK INTERRUPR
   07161 463100002006003                CALL  F,F,LOOP                                ;   CHECK LOOPING
   07162 005152360003400                CA    SUBO,.7,WRK2            POP             ; END
   07163 063100000427150                JMP   T,NZ,B750                               ;

                                 B760:                                                  ; FOR WRK2:= 8 STEP 1 UNTIL 63 DO
   07164 464740363400012                ZABR  ADDO,WRK2,WRK2,DATO                     ; BEGIN
                                                                                        ;   DATAOUT:= WRK2
   07165 421355122403305                CABR  CAND,.1B1,STAT,STAT,CPUST  PUSH         ;   CLEAR ERRORFLAG  , PUSH
   07166 060543174023001 1              ZAR   OR,WRK5,LAD             CONT WRT        ;   LAD:= WRK5  ,WRT
   07167 062473000016200                IZQ   OR,0,710                                ;
   07170 060723003200000                ZQB   OR,WRKO                                 ;   WRKO:= 8'710
```

```
10154 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 07171 065773003300100        XZB   OR,CIOST,WRK1          CONT 0 1    ;   WRK1:= CPUIOSTATUS
03 07172 061354157303400        CAB   AND,.7,WRK1,WRK1                     ;   IF IOSTATUS(21:23)<>0 THEN
04 07173 063100000426163        CALL  T,NZ,TBERR                           ;     TBERR
05 07174 060730203200000        ZBB   ADD0,WRK0                            ;   WRK0:= WRK0+1
06 07175 470523000000000        ZQ    OR                       CONT 0 2    ;   DELAY(900 NSEC)
07 07176 470523000000000        ZQ    OR                       CONT 0 2    ;
08 07177 063100000216002        CALL  F,EXINT,TERROR                       ;   IF -,EXINT THEN
09                                                                         ;     TERROR (711)
10 07200 060523000000022        ZQR   OR,ILEVR                             ;   ILEVR:= 0
11 07201 470523000000000        ZQ    OR                       CONT 0 2    ;   DELAY(675 NSEC)
12 07202 470523000000000        ZQ    OR                       CONT 0 2    ;
13 07203 472473000000760        IZQ   OR,0,377                 CONT 2      ;
14 07204 061764003300300        XQB   AND,ILEV,WRK1                        ;   WRK1= ILEV(16:23)
15 07205 465152354003400        CA    SUB0,.7,WRK1                         ;   IF WRK1=7 THEN
16 07206 063100000427222        JMP   T,NZ,B761                            ;   BEGIN
17 07207 061173000003420        CZR   OR,.7,INTRG                          ;     CLEAR INTERRUPT(7)
18 07210 074523000000000        ZQ    OR                       CONT 0 3    ;     DELAY(900 NSEC)
19 07211 074523000000000        ZQ    OR                       CONT 0 3    ;
20 07212 074523000000000        ZQ    OR                       CONT 0 3    ;
21 07213 074523000000000        ZQ    OR                       CONT 0 3    ;
22 07214 063100000216002        CALL  F,EXINT,TERROR                       ;     IF -,EXINT THEN
23 07215 060523000000022        ZQR   OR,ILEVR                             ;       TERROR
24 07216 470523000000000        ZQ    OR                       CONT 0 2    ;     ILEVR:= 0
25 07217 470523000000000        ZQ    OR                       CONT 0 2    ;     DELAY(675 NSEC)
26 07220 472473000000760        IZQ   OR,0,377                 CONT 2      ;
27 07221 061764003300300        XQB   AND,ILEV,WRK1                        ;     WRK1:= ILEV(16:23)
28                                                                         ;   END
29 07222 060730203200000 B761:  ZBB   ADD0,WRK0                            ;   WRK0:= WRK0+1
30 07223 460512357400000        AB    SUB0,WRK1,WRK2                       ;   IF WRK1<> WRK2 THEN
31 07224 063100000242600Z       CALL  T,NZ,TERROR                          ;     TERROR (712)
32 07225 060730203200000        ZBB   ADD0,WRK0                            ;   WRK0:= WRK0+1
33 07226 460543154000020        ZAR   OR,WRK1,INTRG                        ;   CLEAR INTERRUPT(WRK1)
34 07227 074523000000000        ZQ    OR                       CONT 0 3    ;   DELAY(900 NSEC)
35 07230 074523000000000        ZQ    OR                       CONT 0 3    ;
36 07231 074523000000000        ZQ    OR                       CONT 0 3    ;
37 07232 074523000000000        ZQ    OR                       CONT 0 3    ;
38 07233 063100000261724Z       CALL  T,EXINT,CHINT                        ;   IF EXINT THEN
```

```
IC155 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                                                                          ;     CHECK INTERRUPT (713)
03 C7234 4631000002C06003          CALL  F,F,LOOP                           ;   CHECK LOOPING
04 C7235 0624730000CC1760          IZQ   OR,0,77                            ;
05 C7236 0005023600CCC000          AQ    SUB0,WRK2             POP          ; END
06 C7237 4631000000427164          JMP   T,NZ,B760                          ;
07 C7240 0631000000CC7256          JMP   F,F,B1000                          ; GOTO TEST 1000
08
09
10                         CHINT:                                           ;CHECK INTERRUPT
11 C7241 0605230000CC0022          ZQR   OR,ILEVR                           ; ILEVR:= 0
12 C7242 4705230000CC0000          ZQ    OR              CONT 0 2           ; DELAY(675 NSEC)
13 C7243 4705230000CC0000          ZQ    OR              CONT 0 2           ;
14 C7244 4724730000CC7760          IZQ   OR,0,377        CONT 2             ;
15 C7245 0617640033CC0300          XQB   AND,ILEV,WRK1                      ; WRK1:= ILEV(16:23)
16 C7246 4651523540CC3400          CA    SUB0,.7,WRK1                       ; IF WRK1<> 7 THEN
17 C7247 4631000000426002          JMP   T,NZ,TERROR                        ;   GOTO TERROR
18 C7250 4605431540CC0020          ZAR   OR,WRK1,INTRG                      ; CLEAR INTERRUPT(WRK1)
19 C7251 0745230000CC0000          ZQ    OR              CONT 0 3           ; DELAY(900 NSEC)
20 C7252 0745230000CC0000          ZQ    OR              CONT 0 3           ;
21 C7253 0745230000CC0000          ZQ    OR              CONT 0 3           ;
22 C7254 0745230000CC0000          ZQ    OR              CONT 0 3           ;
23 C7255 6031000000616002          JMP   T,EXINT,TERROR  RTN                ; IF EXINT THEN GOTO TERROR
24                                                                         ; ELSE RETURN
25
26                         ;TEST OF REGISTERS DEFINED BY INSTRUCTIONREGISTER
27                         ;AND TEST OF PREFETCH INSTRUCTIONCOUNTER
28
29 C7256 0613730034CC3000 B1000: CZB   OR,.6,WRK2                          ;
30 C7257 0624730000166340          IZQ   OR,0,B1010/3                       ;
31 C7260 4607230035CC0000          ZQB   OR,WRK3                            ; WRK3:= ADDR(B1010)
32 C7261 4631431643CC0000 B1001: CXZA  OR,WRK3,F,F                         ; FOR WRK2:= 6 STEP -2 UNTIL 0 DO
33 C7262 0605230000CC0012          ZQR   OR,DAT0                            ; BEGIN
34 C7263 0605431600230011          ZAR   OR,WRK2,LAD     CONT WRT           ;   LAD:= GETCONT(WRK3) , WRT
35 C7264 0610730000CC3400          CZQ   OR,.7                              ;
36 C7265 4655640000CC0100          XQ    AND,CIOST       CONT 0 1           ;   IF CPUIOSTATUS(21:23)<>0 THEN
37 C7266 4631000000427347          JMP   T,NZ,B1100                         ;     GOTO B1100  C. NEXT TEST
```

```
10156 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 07267 4607302035CC0000          ZBB   ADDO,WRK3                              ;   WRK3:= WRK3+1
03 07270 4653511634CC3700          CAB   SUN,.1,WRK2,WRK2                       ; END
04 07271 0631000000017261          JMP   F,NEG,B1001                            ;
05 07272 4624730000166560          IZQ   OR,0,B1012/3                           ;
06 07273 0607230037CC0000          ZQB   OR,WRK5                                ; WRK5:= ADDR(B1012)
07 07274 0613730032CC2200          CZB   OR,.1B14,WRK0                          ; WRK0:= 8'1000
08                          B1002:                                              ; REPEAT
09 07275 0653710035CC3700          CZB   SUN,.1,WRK3                            ;   WRK3:= -2
10 07276 0605240000250011          ZQR   AND,LAD                  CONT REAJ     ;   LAD:= 0 , READ TO INSTR REG
11 07277 4607340036CC0000          ZBB   AND,WRK4                               ;   WRK4:= 0
12                          B1003:                                              ;   REPEAT
13 07300 0607431674CC0003          ZABR  OR,WRK3,WRK2,ICD                       ;     WRK2:=ICD:= WRK3
14 07301 7431000000CC7302          JMP   F,F,.+3                  NEXT          ;     LOAD INSTR REG
15 07302 4213551224CC3305          CABR  CAND,.1B1,STAT,STAT,CPUST  PUSH        ;     CLEAR ERROR FLAG
16 07303 0631431743CC0000          CXZA  OR,WRK5,F,F                            ;     IF EXPECTED(WRK5)<>
17 07304 4605123633CC0000          AB    SUBO,WRK2,WRK1                         ;     RECEIVED(WRK4,WRK5) THEN
18 07305 0631000002426002          CALL  T,NZ,TERROR                            ;       TERROR
19 07306 4631000002CC6003          CALL  F,F,LOOP                               ;     TEST LOOPING
20 07307 4053503675CC3700          CAB   ADDO,.1,WRK3,WRK3        POP           ;     WRK3:= WRK3+2
21 07310 0607302032CC0000          ZBB   ADDO,WRK0                              ;     WRK0:= WRK0+1
22 07311 4607302036CC0000          ZBB   ADDO,WRK4                              ;     WRK4:= WRK4+1
23 07312 4651523700CC1500          CA    SUBO,.4,WRK4                           ;   UNTIL WRK4=4
24 07313 4631000000427300          JMP   T,NZ,B1003                             ;
25 07314 4653501777CC1400          CAB   ADD,.3,WRK5,WRK5                       ;   WRK5:= WRK5+3
26 07315 4631000000CC7275          JMP   F,F,B1002                              ; UNTIL FALSE
27
28                          B1010:                                              ;B1010: C. DATA TO BE LOADED IN
29                                                                              ;        INSTRUCTIONSREGISTER
30 07316 2024730014600100          IZQ   OR,0063,0004             RTN           ; ADDR 6
31 07317 2024730010400040          IZQ   OR,0042,0002             RTN           ; ADDR 4
32 07320 6024730004200000          IZQ   OR,0021,0000             RTN           ; ADDR 2
33 07321 2024730000177740          IZQ   OR,0000,7776             RTN           ; ADDR 0
```

```
10157 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          B1011:                                          ;B1011:
03 07322 6007431173CC0000          ZAB   OR,W3,WRK1          RTN           ;
04 07323 6007431033CC0000          ZAB   OR,W0,WRK1          RTN           ;
05 07324 2007431073CC0000          ZAB   OR,W1,WRK1          RTN           ;
06 07325 2007431133CC0000          ZAB   OR,W2,WRK1          RTN           ;
07 07326 6007431173CC0000          ZAB   OR,W3,WRK1          RTN           ;
08
09                          B1012:                                          ;B1012:
10 07327 4607431640CC0000          ZAB   OR,WRK3,W                         ; 1000 - 1003
11 07330 0624730000166460          IZQ   OR,0,B1011/3+1                    ;
12 07331 4631001701CC0000          JXAQ  ADD,WRK4,F,F                      ;
13 07332 0607431641CC0000          ZAB   OR,WRK3,WPRE                      ; 1004 - 1007
14 07333 4624730000166440          IZQ   OR,0,B1011/3                      ;
15 07334 4631001701CC0000          JXAQ  ADD,WRK4,F,F                      ;
16 07335 0607431642CC0000          ZAB   OR,WRK3,X                         ; 1010 - 1013
17 07336 0624730000166460          IZQ   OR,0,B1011/3+1                    ;
18 07337 4631001701CC0000          JXAQ  ADD,WRK4,F,F                      ;
19 07340 2007730033CC0000          RZB   OR,DISP,WRK1        RTN           ; 1014 - 1017
20 07341 0605230000CC0000          ZQ    OR                                ;
21 07342 0605230000CC0000          ZQ    OR                                ;
22 07343 4647501674001000          RAB   ADD,ICS,WRK3,WRK2                 ; 1020 - 1023
23 07344 6007730033CC2000          RZB   OR,DIC,WRK1         RTN           ;
24 07345 0605230000CC0000          ZQ    OR                                ;
25 07346 0005230000CC0000          ZQ    OR                  POP           ;
26
27                          ;TEST OF I/O PROTECTION SYSTEM
28
29                          B1100:                                          ;B1100:
30 07347 4611730000CC3724          CZR   OR,.1,CAMBC                       ; DISABLE CAM
31 07350 4613730020CC3700          CZB   OR,.1,W0                          ; FOR BASE:= 0, 64 DO
32 07351 0605240000CC0013          ZQR   AND,BASE                          ; BEGIN
33 07352 4605240000CC0003          ZQR   AND,ICD                           ;   ICD:= BASE
34 07353 0624730000C22000 B1101:   IZQ   OR,0,1100                        ;   WRK0:= 8'1100
35 07354 0607230032CC0000          ZQB   OR,WRK0                           ;
36 07355 4611730000CC1614          CZR   OR,.16,LLIM                       ;   LLIM:= 16
37 07356 4611730000CC0515          CZR   OR,.22,ULIM                       ;   ULIM:= 22
38 07357 0613730036CC3700          CZB   OR,.1,WRK4                        ;   FOR CPA:= 12,32 DO
39 07360 4611730000CC1116          CZR   OR,.12,CPA                        ;   BEGIN
```

```
10158 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 07361 0624730000170000 B1102: IZQ   OR,0,B1110/3                          ;      FOR LAD:= -2,10,14,20,24,6,4 DO
03 07362 0607230037CC0000        ZQB   OR,WRK5                               ;      FOR FUNC:= READP,WRTP DO
04 07363 4213551224CC3305 B1103: CABR  CAND,.1B1,STAT,STAT,CPUST  PUSH       ;      BEGIN
05 07364 0631431743CC0000        CXZA  OR,WRK5,F,F                           ;        START IO(LAD,FUNC)
06 07365 4624730000147120        IZQ   OR,0,B1111/3-11C0                     ;
07 07366 4631001503CC0000        CXAQ  ADD,WRK0,F,F                          ;        WRK2:= EXPECTED STATUS(WRK0)
08 07367 0607230034CC0000        ZQB   OR,WRK2                               ;
09 07370 4624730000600200        IZQ   OR,3,10                               ;        WRK1:= CPUIOSTAT AND 8'30010
10 07371 0657640033CC0100        XQB   AND,CIOST,WRK1          CONT 0 1      ;
11 07372 4605123574CC0000        AB    SUB0,WRK1,WRK2                        ;        IF WRK1<>WRK2 THEN
12 07373 0631000002426002        CALL  T,NZ,TERROR                           ;          TERROR
13 07374 4631000002CC6003        CALL  F,F,LOOP                              ;        TEST LOOPING
14 07375 0007302032CC0000        ZBB   ADD0,WRK0               POP           ;        WRK0:= WRK0+1
15 07376 0607302037CC0000        ZBB   ADD0,WRK5                             ;
16 07377 4631000000007363        JMP   F,F,B1103                             ;      END FOR FUNC, LAD
17
18                         B1110:                                            ;
19 07400 6051710000203711        CZR   SUN,.1,LAD              RTN READP     ; LAD:= -2  , READP
20 07401 2051710000223711        CZR   SUN,.1,LAD              RTN WRTP      ; LAD:= -2  , WRTP
21 07402 6011730000201311        CZR   OR,.10,LAD              RTN READP     ; LAD:= 10  , READP
22 07403 2011730000221311        CZR   OR,.10,LAD              RTN WRTP      ; LAD:= 10  , WRTP
23 07404 6011730000201011        CZR   OR,.14,LAD              RTN READP     ; LAD:= 14  , READP
24 07405 2011730000221011        CZR   OR,.14,LAD              RTN WRTP      ; LAD:= 14  , WRTP
25 07406 2051720000200611        CZR   SUB,.21,LAD             RTN READP     ; LAD:= 20  , READP
26 07407 6051720000220611        CZR   SUB,.21,LAD             RTN WRTP      ; LAD:= 20  , WRTP
27 07410 6051702000200411        CZR   ADD0,.23,LAD            RTN READP     ; LAD:= 24  , READP
28 07411 2051702000220411        CZR   ADD0,.23,LAD            RTN WRTP      ; LAD:= 24  , WRTP
29 07412 6011730000202011        CZR   OR,.64,LAD              RTN READP     ; LAD:= 64  , READP
30 07413 2011730000222011        CZR   OR,.64,LAD              RTN WRTP      ; LAD:= 64  , WRTP
31 07414 4011730000001716        CZR   OR,.32,CPA              POP           ;
32 07415 4607510036CC0000        ZBB   SUN,WRK4                              ;
33 07416 4631000000027361        JMP   F,NZ,B1102                            ;    END FOR CPA
34 07417 0607310020CC0000        ZBB   SUN,W0                                ;
35 07420 4631000000427425        JMP   T,NZ,B1106                            ;    IF BASE=0 THEN
36 07421 0651712000003712        CZR   SUN0,.1,DATO                          ;      DATO:= -1
37 07422 0611730000002013        CZR   OR,.64,BASE                           ;
38 07423 4611730000002003        CZR   OR,.64,ICD                            ;
39 07424 4631000000007353        JMP   F,F,B1101                             ;
```

```
10159 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 07425 461473000000C0100 B1106: XZQ   OR,CIOST                          ; IF CPUIOSTATUS(21:23)<>0 THEN
03 07426 461164000000C3400         CQ    AND,.7                           ; BEGIN
04 07427 463100000042744 3         JMP   T,NZ,C11EX                       ;
05 07430 062473000002376 0         IZQ   OR,U,1177                        ;   WRKU:= 8'1177 C. TEST RELOCATING
06 07431 060723003200C000 0        ZQB   OR,WRKU                          ;
07 07432 462473000000C250 0        IZQ   OR,U,84.                         ;
08 07433 060523000021001 1         ZQR   OR,LAD              CONT READ    ;   LAD:= 84  , READ
09 07434 071570200000C400 0        XZ    ADDO,DATI           CONT U 2     ;   IF DATI<>-1 THEN
10 07435 063100000242600 2         CALL  T,NZ,TERROR                      ;     TERROR
11 07436 020524000000C001 2        ZQR   AND,DATO            PUSH         ; DATO:= U ,  PUSH
12 07437 065172000022061 1         CZR   SUB,.21,LAD         CONT WRTP    ;   LAD:= 20  , WRTP
13 07440 465573000000C010 0        XZ    OR,CIOST            CONT U 1     ;
14 07441 463100000200600 3         CALL  F,F,LOOP                         ;   TESTLOOPING
15 07442 000523000000C000 0        ZQ    OR                  POP          ; END
16 07443 460524000000C002 4 C11EX: ZQR   AND,CAMBC                        ; ENABLE CAM
17 07444 063100000000616 7         JMP   F,F,CPULP                        ; GOTO TEST CPURUNS
```

```

                              B1111:                                                            ;B1111:
                                                                                                ; C. TABLE OF EXPECTED VALUES OF IOSTATUS
C7445 20247300000000200          IZQ    OR,0,10                          RTN                   ; 1100: LAD<U<CPA<LL<UL   , READP
C7446 20247300000000200          IZQ    OR,0,10                          RTN                   ; 1101: LAD<U<CPA<LL<UL   , WRTP
C7447 60247300000600000          IZQ    OR,3,0                           RTN                   ; 1102: U<LAD<CPA<LL<UL   , READP
C7450 20247300000600200          IZQ    OR,3,10                          RTN                   ; 1103: U<LAD<CPA<LL<UL   , WRTP
C7451 20247300000600200          IZQ    OR,3,10                          RTN                   ; 1104: U<CPA<LAD<LL<UL   , READP
C7452 20247300000600200          IZQ    OR,3,10                          RTN                   ; 1105: U<CPA<LAD<LL<UL   , WRTP
C7453 20247300000400000          IZQ    OR,2,0                           RTN                   ; 1106: U<CPA<LL<LAD<UL   , READP
C7454 20247300000400000          IZQ    OR,2,0                           RTN                   ; 1107: U<CPA<LL<LAD<UL   , WRTP
C7455 20247300000000200          IZQ    OR,0,10                          RTN                   ; 1110: U<CPA<LL<UL<LAD   , READP
C7456 20247300000000200          IZQ    OR,0,10                          RTN                   ; 1111: U<CPA<LL<UL<LAD   , WRTP
C7457 20247300000000200          IZQ    OR,0,10                          RTN                   ; 1112: U<CPA<LL<UL<LAD   , READP
C7460 20247300000000200          IZQ    OR,0,10                          RTN                   ; 1113: U<CPA<LL<UL<LAD   , WRTP
C7461 20247300000000200          IZQ    OR,0,10                          RTN                   ; 1114: LAD<U<LL<UL<CPA   , READP
C7462 20247300000000200          IZQ    OR,0,10                          RTN                   ; 1115: LAD<U<LL<UL<CPA   , WRTP
C7463 60247300000600000          IZQ    OR,3,0                           RTN                   ; 1116: U<LAD<LL<UL<CPA   , READP
C7464 20247300000600200          IZQ    OR,3,10                          RTN                   ; 1117: U<LAD<LL<UL<CPA   , WRTP
C7465 60247300000600000          IZQ    OR,3,0                           RTN                   ; 1120: U<LAD<LL<UL<CPA   , READP
C7466 20247300000600200          IZQ    OR,3,10                          RTN                   ; 1121: U<LAD<LL<UL<CPA   , WRTP
C7467 20247300000400000          IZQ    OR,2,0                           RTN                   ; 1122: U<LL<LAD<UL<CPA   , READP
C7470 20247300000400000          IZQ    OR,2,0                           RTN                   ; 1123: U<LL<LAD<UL<CPA   , WRTP
C7471 60247300000000000          IZQ    OR,0,0                           RTN                   ; 1124: U<LL<UL<LAD<CPA   , READP
C7472 20247300000000200          IZQ    OR,0,10                          RTN                   ; 1125: U<LL<UL<LAD<CPA   , WRTP
C7473 20247300000000200          IZQ    OR,0,10                          RTN                   ; 1126: U<LL<UL<CPA<LAD   , READP
C7474 20247300000000200          IZQ    OR,0,10                          RTN                   ; 1127: U<LL<UL<CPA<LAD   , WRTP
```

```
10161 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;CONTENTS OF REGISTERS DURING MEMORY TEST
03
04                          ;WRK0:   STATUS
05                          ;WRK1:   RECEIVED DATA
06                          ;WRK2:   EXPECTED DATA
07                          ;WRK3:   NEWDATA
08                          ;WRK4:   WORK
09                          ;WRK5:   7
10                          ;SB:     OLDDATA
11                          ;W0:     CURRENT MEMORY START
12                          ;W1:     CURRENT INCREMENT
13                          ;W3:     LAST MEMORY LOCATION
14                          ;ICS:    MEMORY ADDRESS
15
16                          MEMTEST:                                          ;MEMTEST START
17 07475 4613531471003406           CABR  OR,.7,CNTR,CNTR,CNTR0               ;
18 07476 0607151431000006           ABBR  CAND,.2,CNTR,CNTR0                  ; TURN LAMP 2 ON
19 07477 4624730000000160           IZQ   OR,0,7                              ; IF TCP THEN
20 07500 0631000002164013           CALL  F,MMODE,TPTXT                       ;   TYPE(BELL)
21 07501 4213551224003305           CABR  CAND,.1B1,STAT,STAT,CPUST PUSH      ; CLEAR ERRORFLAG  , PUSH
22 07502 4611730000003724           CZR   OR,.1,CAMBC                         ; CAMBC:= 1 C. DISABLE CAM
23 07503 0631000002006004           CALL  F,F,SIZEM                           ; SIZE AND CLEAR MEM
24 07504 4631000002016163           CALL  F,NEG,TBERR                         ; IF IOERROR THEN TBERR
25 07505 4631000002006003           CALL  F,F,LOOP                            ; TEST LOOPING
26 07506 0007340026000000           ZBB   AND,SB                    POP       ;
27 07507 4607340035000000           ZBB   AND,WRK3                            ; OLDDATA:= NEWDATA:= 0
28 07510 0653702021003700           CZB   ADD0,.1,W1                          ; W1:= 2 DISPLACEMENT
29 07511 4613730037003400           CZB   OR,.7,WRK5                          ; WRK5:= 7
30 07512 0651503140003715           CAR   ADD0,.1,W3,ULIM                     ;
31 07513 0651503140003716           CAR   ADD0,.1,W3,CPA                      ; ULIM:=CPA:= W3+2
32 07514 0605240000000013           ZQR   AND,BASE                            ; BASE:= 0
33 07515 0611730000003614           CZR   OR,.8,LLIM                          ; LLIM:= 8
34                          A02:                                              ; WHILE W1< MEMSIZE DO
35 07516 0613551224002405           CABR  CAND,.1B7,STAT,STAT,CPUST           ; BEGIN
```

```
10162 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          A03:                                                    ;   FOR DOWN:= FALSE,TRUE DO
03 07517 4653702025CC3700           CZB    ADDO,.1,CAUSE                            ;   BEGIN
04 07520 4611541200CC3100           CA     AND,.1B3,STAT                            ;     FOR CAUSE:=0 STEP 1 UNTIL
05 07521 0631000000C27524           JMP    F,NZ,A04                                 ;     IF SHORT THEN 1 ELSE .47 DO
06 07522 0624730000001400           IZQ    OR,0,48.                                 ;     BEGIN
07 07523 0607230025C00000           ZQB    OR,CAUSE                                 ;
08 07524 4631000002CC7632 A04:      CALL   F,F,GETDATA                              ;       NEWDATA:=GETDATA(CAUSE)
09 07525 0611541200CC2400           CA     AND,.1B7,STAT                            ;       IF -,DOWN THEN
10 07526 0631000000427562           JMP    T,NZ,DOWN                                ;       BEGIN
11 07527 0611730000003603           CZR    OR,.8,ICD                                ;         FOR W0:=8 STEP 2 UNTILDISP-2 DO
12 07530 4607730020001000 A10:      RZB    OR,ICS,W0                                ;         FOR ICS:=W0
13                                                                                   ;               STEP DISP UNTIL LAST DO
14 07531 4213551224CC3305 A20:      CABR   CAND,.1B1,STAT,STAT,CPUST PUSH           ;         BEGIN
15 07532 0605730000201011           RZR    OR,ICS,LAD                CONT READP     ;
16 07533 0645513140CC1000           RA     SUNO,ICS,W3                              ;
17 07534 4631000000417555           JMP    T,NEG,A30                                ;           DATI:= MEM(IC)
18 07535 0605431640CC0012           ZAR    OR,WRK3,DATO                             ;           IF IOERROR THEN
19 07536 0655541740CC0100           XA     AND,CIOST,WRK5            CONT U 1       ;             IBERROR
20 07537 0631000002426163           CALL   T,NZ,IBERR                               ;
21 07540 4655523300004000           XA     SUBO,DATI,SB                             ;           IF DATI<>OLDDATA THEN
22 07541 4631000002427652           CALL   T,NZ,ODERR                               ;             OLDDATAERROR;
23 07542 4605730000221011           RZR    OR,ICS,LAD                CONT WRTP      ;           MEM(IC):= NEWDATA
24 07543 0655541740CC0100           XA     AND,CIOST,WRK5            CONT U 1       ;           IF IOERROR THEN
25 07544 0631000002426163           CALL   T,NZ,IBERR                               ;             IBERR;
26 07545 0605730002201011           RZR    OR,ICS,LAD                CONT READP     ;           DATI:= MEM(IC)
27 07546 0655541740CC0100           XA     AND,CIOST,WRK5            CONT U 1       ;           IF IOERROR THEN
28 07547 0631000002426163           CALL   T,NZ,IBERR                               ;             IBERR
29 07550 0655523640CC4000           XA     SUBO,DATI,WRK3                           ;           IF DATI<>NEWDATA THEN
30 07551 4631000002427646           CALL   T,NZ,NDERR                               ;             NEWDATAERROR
31 07552 4631000002CC6003           CALL   F,F,LOOP                                 ;           TEST LOOPING
32 07553 4045501040CC1003           RAR    ADD,ICS,W1,ICD            POP            ;         END FOR ICS
33 07554 0631000000CC7531           JMP    F,F,A20                                  ;
34                          A30:                                                    ;
35 07555 4651503000CC3703           CAR    ADDO,.1,W0,ICD                           ;         END FOR W0
36 07556 0650513000003000           CAQ    SUNO,.6,W0                               ;
37 07557 4605013040CC0000           AQ     SUNO,W1                                  ;
38 07560 4631000000417530           JMP    T,NEG,A10                                ;
39 07561 4631000000CC7614           JMP    F,F,A40                                  ;         OLDDATA:= NEWDATA
```

```
10163 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          DOWN:                                                ;         END  IF -,DOWN
03 07562 4605431140000003           ZAR    OR,W3,ICD                             ;         ELSE
04                                                                               ;         BEGIN   DOWN
05 07563 4607730020001000 A101:     RZB    OR,ICS,W0                             ;           FOR W0:= LAST STEP -2
06                                                                               ;              UNTIL LAST-DISP+2 DO
07 07564 4213551224003305 A201:     CABR   CAND,.1B1,STAT,STAT,CPUST PUSH        ;           FOR ICS:= W0 STEP -DISP UNTIL 8
08 07565 4607730036201011           RZBR   OR,ICS,WRK4,LAD          CONT READP   ;           BEGIN
09 07566 0651513700003600           CA     SUN0,.8,WRK4                          ;             DATI:= MEM(IC)
10 07567 0631000000417610           JMP    T,NEG,A301                            ;             IF IOERROR THEN
11 07570 0605431640000012           ZAR    OR,WRK3,DATO                          ;                 TBERR
12 07571 0655541740000100           XA     AND,CIOST,WRK5           CONT 0 1     ;
13 07572 0631000002426163           CALL   T,NZ,TBERR                            ;
14 07573 4655523300004000           XA     SUB0,DATI,SB                          ;             IF DATI<>OLDDATA THEN
15 07574 4631000002427652           CALL   T,NZ,ODERR                            ;               OLDDATAERROR;
16 07575 4605730000221011           RZR    OR,ICS,LAD               CONT WRTP    ;             MEM(IC):= NEWDATA
17 07576 0655541740000100           XA     AND,CIOST,WRK5           CONT 0 1     ;             IF IOERROR THEN
18 07577 0631000002426163           CALL   T,NZ,TBERR                            ;               TBERR;
19 07600 0605730000201011           RZR    OR,ICS,LAD               CONT READP   ;             DATI:= MEM(IC)
20 07601 0655541740000100           XA     AND,CIOST,WRK5           CONT 0 1     ;             IF IOERROR THEN
21 07602 0631000002426163           CALL   T,NZ,TBERR                            ;               TBERR
22 07603 0655523640004000           XA     SUB0,DATI,WRK3                        ;             IF DATI<>NEWDATA THEN
23 07604 4631000002427646           CALL   T,NZ,NDERR                            ;               NEWDATAERROR
24 07605 4631000002006003           CALL   F,F,LOOP                              ;             TEST LOOPING
25 07606 4045523040001003           RAR    SUB0,ICS,W1,ICD          POP          ;
26 07607 4631000000017564           JMP    F,NEG,A201                            ;           END  FOR ICS
27 07610 4651511000003703 A301:     CAR    SUN,.1,W0,ICD                         ;
28 07611 4644513140001000           RAQ    SUN0,ICS,W3                           ;
29 07612 4605013040000000           AQ     SUN0,W1                               ;
30 07613 4631000000417563           JMP    T,NEG,A101                            ;           END  FOR W0
31 07614 0607431666000000 A40:      ZAB    OR,WRK3,SB                            ;           OLDDATA:= NEWDATA
32                                                                               ;         END  DOWN
33 07615 0607310025000000           ZBB    SUN,CAUSE                             ;
34 07616 4631000000427524           JMP    T,NZ,A04                              ;       END  FOR CAUSE
35 07617 0613561224002405           CABR   EXOR,.1B7,STAT,STAT,CPUST             ;
36 07620 0611541200002400           CA     AND,.1B7,STAT                         ;
37 07621 4631000000427517           JMP    T,NZ,A03                              ;   END   FOR DOWN
```

```
10164 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02 07622 0637330021CC0060        SLZBB OR,W1,Z,NL,F,F                    ;   W1:= IF FULL TEST THEN W1 SHIFT 1
03 07623 4611541200C03100        CA    AND,.1B3,STAT                     ;   ELSE W1 SHIFT 2
04 07624 0631000000427626        JMP   T,NZ,A41                          ;
05 07625 0637330021CC0060        SLZBB OR,W1,Z,NL,F,F                    ;
06 07626 4605123161CC0000 A41:   AB    SUBO,W3,W1                        ;
07 07627 4631000000017516        JMP   F,NEG,AU2                         ; END   WHILE DISP< MEMSIZE
08 07630 4605240000CC0024        ZQR   AND,CAMBC                         ; CAMBYPASS:= 0
09 07631 4631000000C06075        JMP   F,F,TU.3                          ;
10
11                         GETDATA:                                      ;GETDATA
12 07632 4611541200C03100        CA    AND,.1B3,STAT                     ;
13 07633 4631000000427641        JMP   T,NZ,EXTDATA                      ; IF -,FULL TEST THEN
14 07634 0624731252525240        IZQ   OR,5252,5252                      ; BEGIN
15 07635 4607230035CC0000        ZQB   OR,WRK3                           ;   NEWDATA:= 8'52525252
16 07636 0605411240CC0000        ZA    SUN,CAUSE                         ;   IF CAUSE=1 THEN
17 07637 2031000000C27640        JMP   F,NZ,.+3              RTN         ;     NEWDATA:= -,NEWDATA
18 07640 6007421675CC0000        ZAB   SUB,WRK3,WRK3         RTN         ; END ELSE
19                         EXTDATA:                                      ; BEGIN
20 07641 0637330035CC0060        SLZBB OR,WRK3,Z,NL,F,F                  ;   NEWDATA:= NEWDATA SHIFT -1
21 07642 0624730000CC0600        IZQ   OR,U,24.                          ;   IF CAUSE>24 THEN
22 07643 0605013240CC0000        AQ    SUNO,CAUSE                        ;     CAUSE:= CAUSE+1
23 07644 6031000000417645        JMP   T,NEG,.+3             RTN         ; END
24 07645 2007302035CC0000        ZBB   ADDO,WRK3             RTN         ;
25
26                         NDERR:                                        ;NEWDATAERROR
27 07646 4617730033CC4000        XZB   OR,DATI,WRK1                      ; WRK1:= DATI
28 07647 0607340032CC0000        ZBB   AND,WRKU                          ; WRKU:= U
29 07650 0607431674CC0000        ZAB   OR,WRK3,WRK2                      ; WRK2:= WRK3 EXPECTED:= NEWDATA
30 07651 4631000000CC6002        JMP   F,F,TERROR                        ; GOTU TERROR
31
32                         ODERR:                                        ;OLDDATAERROR
33 07652 4617730033CC4000        XZB   OR,DATI,WRK1                      ; WRK1:= DATI
34 07653 4613730032CC3700        CZB   OR,.1,WRKU                        ; WRKU:= 1
35 07654 4607431334CC0000        ZAB   OR,SB,WRK2                        ; WRK2:= SB EXPECTED:= OLDDATA
36 07655 4631000000CC6002        JMP   F,F,TERROR                        ; GOTU TERROR
37
```

```
10165 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
02                          ;SIZEMEM:                                          ;PROCEDURE TO CLEAR AND SIZE MEMORY
03                          .IZEMEM:                                           ;
04                                                                             ; CONDITIONS AT RETURN:
05                                                                             ;   NEG=T  C. OK
06                                                                             ;   NEG=F  C. IOERROR
07                                                                             ; BEGIN
08 07656 4613730036003000           CZB   OR,.6,WRK4                         ;   WRK4:= 6
09                                                                             ;   REPEAT
10 07657 4653503736003712 SIZE1:    CABR  ADDO,.1,WRK4,WRK4,DATO             ;     DATO:= WRK4:= WRK4+2
11 07660 4605431700230011           ZAR   OR,WRK4,LAD             CONT WRT   ;     LAD:= WRK4  ,WRT
12 07661 0610730000003400           CZQ   OR,.7                              ;     Q:= 7
13 07662 4655640000000100           XQ    AND,CIOST               CONT 0 1   ;     IF IOERROR THEN
14 07663 4631000000427671           JMP   T,NZ,ADREND                        ;       GOTO ADREND
15 07664 0605431700210011           ZAR   OR,WRK4,LAD             CONT READ  ;     LAD:= WRK4  ,READ
16 07665 0655640000004100           XQ    AND,IOSTA               CONT 0 1   ;     IF IOERROR THEN
17 07666 4631000000427671           JMP   T,NZ,ADREND                        ;       GOTO ADREND
18 07667 0655523700004000           XA    SUBO,DATI,WRK4                     ;     IF DATAIN<>WRK4 THEN
19 07670 4631000000027657           JMP   F,NZ,SIZE1                         ;       GOTO ADREND
20                                                                             ;   UNTIL FALSE
21                          ADREND:                                            ;ADREND:
22 07671 4605240000000012           ZQR   AND,DATO                           ;   DATAOUT:= 0
23 07672 4653511723233711           CABR  SUN,.1,WRK4,W3,LAD      CONT WRT   ;   LAD:=W3:= WRK4-2 C. LAST MEM LOC
24 07673 0605431700000003 ADR1:     ZAR   OR,WRK4,ICD                        ;
25 07674 4607113436000000           ABB   SUNO,.2,WRK4                       ;
26 07675 4631000000417702           JMP   T,NEG,ADR2                         ;
27 07676 4655640000000100           XQ    AND,CIOST               CONT 0 1   ;   FOR WRK4:= W3,WRK4-2 WHILE
28 07677 2031000000027700           JMP   F,NZ,.+3                RTN        ;    WRK4>=0 AND -,IOERROR DO
29 07700 4605431700230011           ZAR   OR,WRK4,LAD             CONT WRT   ;     LAD:= WRK4 ,  WRT
30 07701 0631000000007673           JMP   F,F,ADR1                           ;
31 07702 4655640000000100 ADR2:     XQ    AND,CIOST               CONT 0 1   ;
32 07703 6051712000003712           CZR   SUNO,.1,DATO            RTN        ;
33
```

```
10166 CPU823 MICROPROGRAM LBJ 850816 VERSION 8
01
0000 SOURCE LINES IN ERROR
```

```
 0001 .MAIN DOMUS MACRO ASSEMBLER REV 02.00
01
02                ; INSTRUCTION EXECUTION DECODING MAP
03
04                ; 0: ESCAPE MODE
05                ; 1: FPU NOT INSTALLED
06                ; 2: INSTR BIT(6)
07                ; 3:          (7)
08                ; 4: INSTR BIT(0)
09                ; 5:          (1)
10                ; 6:          (2)
11                ; 7:          (3)
12                ; 8:          (4)
13                ; 9:          (5)
14
```

```
10002 .MAIN
01        000000 .LOC 0
02
03 00000 000032 INTO/3                    ;FPU INSTALLED
04 00001 002274 DO/3
05 00002 000250 EL/3
06 00003 000227 HL/3
07 00004 000073 LA/3
08 00005 000100 LO/3
09 00006 000105 LX/3
10 00007 000112 WA/3
11 00010 000121 WS/3
12 00011 000053 AM/3
13 00012 000145 WMFPU/3
14 00013 000057 AL/3
15 00014 002472 XI/3
16 00015 000624 JL/3
17 00016 000637 JD/3
18 00017 000674 JE/3
19 00020 000304 XL/3
20 00021 000272 ES/3
21 00022 000260 EA/3
22 00023 000240 ZL/3
23 00024 000066 RL/3
24 00025 002245 SP/3
25 00026 001351 RE/3
26 00027 000316 RS/3
27 00030 000165 WD/3
28 00031 000412 RX/3
29 00032 000360 HS/3
30 00033 000376 XS/3
31 00034 005225 GG/3
32 00035 002257 DI/3
33 00036 002403 AP/3
34 00037 002367 UL/3
```

```
!0CC3 .MAIN
C1
02 CC040 000711 C1/3
03 CC041 000062 AC/3
04 CC042 000562 NS/3
05 CC043 000606 ND/3
06 CC044 000501 AS/3
07 CC045 000513 AD/3
08 CC046 000533 LS/3
09 CC047 000545 LD/3
10 CC050 002200 SH/3
11 CC051 002205 SL/3
12 CC052 002212 SE/3
13 CC053 002217 SN/3
14 CC054 002224 SO/3
15 CC055 002232 SZERO/3
16 CC056 002237 SX/3
17 CC057 005232 GP/3
18 CC060 001270 FAFPU/3
19 CC061 001343 FSFPU/3
20 CC062 001066 FMFPU/3
21 CC063 000032 INTC/3
22 CC064 001213 FDFPU/3
23 CC065 000757 CF/3
24 CC066 000431 DL/3
25 CC067 000332 DS/3
26 CC070 000443 AA/3
27 CC071 000462 SS/3
28 CC072 002447 DP/3
29 CC073 003010 MH/3
30 CC074 002316 LK/3
31 CC075 003044 IX/3
32 CC076 000032 INTC/3
33 CC077 000032 INTC/3
```

```
!0004 .MAIN
01
02        000005 .DO 3              ; W=FIELD<>0
03 C0100  000032 INTG/3
04 C0101  002274 DO/3
05 C0102  000250 EL/3
06 C0103  000227 HL/3
07 C0104  000073 LA/3
08 C0105  000100 LO/3
09 C0106  000105 LX/3
10 C0107  000112 WA/3
11 C0110  000121 WS/3
12 C0111  000053 AM/3
13 C0112  000145 WMFPU/3
14 C0113  000057 AL/3
15 C0114  002472 XI/3
16 C0115  000631 JLL/3
17 C0116  000657 JDL/3
18 C0117  000702 JEL/3
19 C0120  000304 XL/3
20 C0121  000272 ES/3
21 C0122  000260 EA/3
22 C0123  000240 ZL/3
23 C0124  000066 RL/3
24 C0125  002245 SP/3
25 C0126  001351 RE/3
26 C0127  000316 RS/3
27 C0130  000165 WD/3
28 C0131  000412 RX/3
29 C0132  000360 HS/3
30 C0133  000376 XS/3
31 C0134  005225 GG/3
32 C0135  002257 DI/3
33 C0136  002403 AP/3
34 C0137  002367 UL/3
```

```
10005 .MAIN
01
02 00140 000711 CI/3
03 00141 000062 AC/3
04 00142 000562 NS/3
05 00143 000606 ND/3
06 00144 000501 AS/3
07 00145 000513 AD/3
08 00146 000533 LS/3
09 00147 000545 LD/3
10 00150 002200 SH/3
11 00151 002205 SL/3
12 00152 002212 SE/3
13 00153 002217 SN/3
14 00154 002224 SO/3
15 00155 002232 SZERO/3
16 00156 002237 SX/3
17 00157 005232 GP/3
18 00160 001270 FAFPU/3
19 00161 001343 FSFPU/3
20 00162 001066 FMFPU/3
21 00163 000032 INTC/3
22 00164 001213 FDFPU/3
23 00165 000757 CF/3
24 00166 000431 DL/3
25 00167 000332 DS/3
26 00170 000443 AA/3
27 00171 000462 SS/3
28 00172 002447 DP/3
29 00173 003010 MH/3
30 00174 002316 LK/3
31 00175 003044 IX/3
32 00176 000032 INTC/3
33 00177 000032 INTC/3
```

```
10006 .MAIN
01
02                .ENDC
03 00200 000032 INTC/3
04 00201 002274 DO/3
05 00202 000250 EL/3
06 00203 000227 HL/3
07 00204 000073 LA/3
08 00205 000100 LO/3
09 00206 000105 LX/3
10 00207 000112 WA/3
11 00210 000121 WS/3
12 00211 000053 AM/3
13 00212 000145 WMFPU/3
14 00213 000057 AL/3
15 00214 002472 XI/3
16 00215 000651 JLL/3
17 00216 000657 JDL/3
18 00217 000702 JEL/3
19 00220 000304 XL/3
20 00221 000272 ES/3
21 00222 000260 EA/3
22 00223 000240 ZL/3
23 00224 000066 RL/3
24 00225 002245 SP/3
25 00226 001351 RE/3
26 00227 000316 RS/3
27 00230 000165 WD/3
28 00231 000412 RX/3
29 00232 000360 HS/3
30 00233 000376 XS/3
31 00234 005225 GG/3
32 00235 002257 DI/3
33 00236 002403 AP/3
34 00237 002367 UL/3
```

```
10007 .MAIN
01
02 00240 000711 CI/3
03 00241 000062 AC/3
04 00242 000562 NS/3
05 00243 000606 ND/3
06 00244 000501 AS/3
07 00245 000513 AD/3
08 00246 000533 LS/3
09 00247 000545 LD/3
10 00250 002200 SH/3
11 00251 002205 SL/3
12 00252 002212 SE/3
13 00253 002217 SN/3
14 00254 002224 SO/3
15 00255 002232 SZERO/3
16 00256 002237 SX/3
17 00257 005232 GP/3
18 00260 001270 FAFPU/3
19 00261 001343 FSFPU/3
20 00262 001066 FMFPU/3
21 00263 000032 INTC/3
22 00264 001213 FDFPU/3
23 00265 000757 CF/3
24 00266 000431 DL/3
25 00267 000332 DS/3
26 00270 000443 AA/3
27 00271 000462 SS/3
28 00272 002447 DP/3
29 00273 003010 MH/3
30 00274 002316 LK/3
31 00275 003044 IX/3
32 00276 000032 INTC/3
33 00277 000032 INTC/3
```

```
10008 .MAIN
01
02                .ENDC
03 00300 000032 INTC/3
04 00301 002274 DO/3
05 00302 000250 EL/3
06 00303 000227 HL/3
07 00304 000073 LA/3
08 00305 000100 LO/3
09 00306 000105 LX/3
10 00307 000112 WA/3
11 00310 000121 WS/3
12 00311 000053 AM/3
13 00312 000145 WMFPU/3
14 00313 000057 AL/3
15 00314 002472 XI/3
16 00315 000631 JLL/3
17 00316 000657 JDL/3
18 00317 000702 JEL/3
19 00320 000304 XL/3
20 00321 000272 ES/3
21 00322 000260 EA/3
22 00323 000240 ZL/3
23 00324 000066 RL/3
24 00325 002245 SP/3
25 00326 001351 RE/3
26 00327 000316 RS/3
27 00330 000165 WD/3
28 00331 000412 RX/3
29 00332 000360 HS/3
30 00333 000376 XS/3
31 00334 005225 GG/3
32 00335 002257 D1/3
33 00336 002403 AP/3
34 00337 002367 UL/3
```

```
10009 .MAIN
01
02 00340 000711 CI/3
03 00341 000062 AC/3
04 00342 000562 NS/3
05 00343 000606 ND/3
06 00344 000501 AS/3
07 00345 000513 AD/3
08 00346 000533 LS/3
09 00347 000545 LD/3
10 00350 002200 SH/3
11 00351 002205 SL/3
12 00352 002212 SE/3
13 00353 002217 SN/3
14 00354 002224 SO/3
15 00355 002232 SZERO/3
16 00356 002237 SX/3
17 00357 005232 GP/3
18 00360 001270 FAFPU/3
19 00361 001343 FSFPU/3
20 00362 001066 FMFPU/3
21 00363 000032 INTC/3
22 00364 001213 FDFPU/3
23 00365 000757 CF/3
24 00366 000431 DL/3
25 00367 000332 DS/3
26 00370 000443 AA/3
27 00371 000462 SS/3
28 00372 002447 DP/3
29 00373 003010 MH/3
30 00374 002316 LK/3
31 00375 003044 IX/3
32 00376 000032 INTC/3
33 00377 000032 INTC/3
```

```
1001C .MAIN
01
02                .ENDC
03 CC400 000032 INTC/3                  ;FPU NOT INSTALLED
04 CC401 002274 DO/3
05 CC402 000250 EL/3
06 CC403 000227 HL/3
07 CC404 000073 LA/3
08 CC405 000100 LO/3
09 CC406 000105 LX/3
10 CC407 000112 WA/3
11 CC410 000121 WS/3
12 CC411 000053 AM/3
13 CC412 000130 WM/3
14 CC413 000057 AL/3
15 CC414 002472 XI/3
16 CC415 000624 JL/3
17 CC416 000637 JD/3
18 CC417 000674 JE/3
19 CC420 000304 XL/3
20 CC421 000272 ES/3
21 CC422 000260 EA/3
22 CC423 000240 ZL/3
23 CC424 000066 RL/3
24 CC425 002245 SP/3
25 CC426 001351 RE/3
26 CC427 000316 RS/3
27 CC430 000165 WD/3
28 CC431 000412 RX/3
29 CC432 000360 HS/3
30 CC433 000376 XS/3
31 CC434 005225 GG/3
32 CC435 002257 DI/3
33 CC436 002403 AP/3
34 CC437 002367 UL/3
```

```
!0011 .MAIN
01
02 CC440 000711 CI/3
03 CC441 000062 AC/3
04 CC442 000562 NS/3
05 CC443 000606 ND/3
06 CC444 000501 AS/3
07 CC445 000513 AD/3
08 CC446 000533 LS/3
09 CC447 000545 LD/3
10 CC450 002200 SH/3
11 CC451 002205 SL/3
12 CC452 002212 SE/3
13 CC453 002217 SN/3
14 CC454 002224 SO/3
15 CC455 002232 SZERO/3
16 CC456 002237 SX/3
17 CC457 005232 GP/3
18 CC460 001221 FA/3
19 CC461 001276 FS/3
20 CC462 001010 FM/3
21 CC463 000032 INTO/3
22 CC464 001135 FD/3
23 CC465 000757 CF/3
24 CC466 000431 DL/3
25 CC467 000332 DS/3
26 CC470 000443 AA/3
27 CC471 000462 SS/3
28 CC472 002447 DP/3
29 CC473 003010 MH/3
30 CC474 002316 LK/3
31 CC475 003044 IX/3
32 CC476 000032 INTO/3
33 CC477 000032 INTO/3
```

```
10012 .MAIN
01
02        000003 .DO 3              ; W=FIELD<>0
03 CC500  000032 INTO/3
04 CC501  002274 DO/3
05 CC502  000250 EL/3
06 CC503  000227 HL/3
07 CC504  000073 LA/3
08 CC505  000100 LO/3
09 CC506  000105 LX/3
10 CC507  000112 WA/3
11 CC510  000121 WS/3
12 CC511  000053 AM/3
13 CC512  000130 WM/3
14 CC513  000057 AL/3
15 CC514  002472 XI/3
16 CC515  000651 JLL/3
17 CC516  000657 JDL/3
18 CC517  000702 JEL/3
19 CC520  000304 XL/3
20 CC521  000272 ES/3
21 CC522  000260 EA/3
22 CC523  000240 ZL/3
23 CC524  000066 RL/3
24 CC525  002245 SP/3
25 CC526  001351 RE/3
26 CC527  000316 RS/3
27 CC530  000165 WD/3
28 CC531  000412 RX/3
29 CC532  000360 HS/3
30 CC533  000376 XS/3
31 CC534  005225 GG/3
32 CC535  002257 DI/3
33 CC536  002403 AP/3
34 CC537  002367 UL/3
```

```
!0013 .MAIN
01
02 C0540 000711 CI/3
03 C0541 000062 AC/3
04 C0542 000562 NS/3
05 C0543 000606 ND/3
06 C0544 000501 AS/3
07 C0545 000513 AD/3
08 C0546 000533 LS/3
09 C0547 000545 LD/3
10 C0550 002200 SH/3
11 C0551 002205 SL/3
12 C0552 002212 SE/3
13 C0553 002217 SN/3
14 C0554 002224 SO/3
15 C0555 002232 SZERO/3
16 C0556 002237 SX/3
17 C0557 005232 GP/3
18 C0560 001221 FA/3
19 C0561 001276 FS/3
20 C0562 001010 FM/3
21 C0563 000032 INTC/3
22 C0564 001135 FD/3
23 C0565 000757 CF/3
24 C0566 000431 DL/3
25 C0567 000332 DS/3
26 C0570 000443 AA/3
27 C0571 000462 SS/3
28 C0572 002447 DP/3
29 C0573 003010 MH/3
30 C0574 002316 LK/3
31 C0575 003044 IX/3
32 C0576 000032 INTC/3
33 C0577 000032 INTC/3
```

```
10014 .MAIN
01
02                .ENDC
03 CC600 000032 INT0/3
04 CC601 002274 DO/3
05 CC602 000250 EL/3
06 CC603 000227 HL/3
07 CC604 000073 LA/3
08 CC605 000100 LO/3
09 CC606 000105 LX/3
10 CC607 000112 WA/3
11 CC610 000121 WS/3
12 CC611 000053 AM/3
13 CC612 000130 WM/3
14 CC613 000057 AL/3
15 CC614 002472 XI/3
16 CC615 000631 JLL/3
17 CC616 000657 JDL/3
18 CC617 000702 JEL/3
19 CC620 000304 XL/3
20 CC621 000272 ES/3
21 CC622 000260 EA/3
22 CC623 000240 ZL/3
23 CC624 000066 RL/3
24 CC625 002245 SP/3
25 CC626 001351 RE/3
26 CC627 000316 RS/3
27 CC630 000165 WD/3
28 CC631 000412 RX/3
29 CC632 000360 HS/3
30 CC633 000376 XS/3
31 CC634 005225 GG/3
32 CC635 002257 DI/3
33 CC636 002403 AP/3
34 CC637 002367 UL/3
```

```
10015 .MAIN
01
02 CC640 000711 CI/3
03 CC641 000062 AC/3
04 CC642 000562 NS/3
05 CC643 000606 ND/3
06 CC644 000501 AS/3
07 CC645 000513 AD/3
08 CC646 000535 LS/3
09 CC647 000545 LD/3
10 CC650 002200 SH/3
11 CC651 002205 SL/3
12 CC652 002212 SE/3
13 CC653 002217 SN/3
14 CC654 002224 SO/3
15 CC655 002232 SZERO/3
16 CC656 002237 SX/3
17 CC657 005232 GP/3
18 CC660 001221 FA/3
19 CC661 001276 FS/3
20 CC662 001010 FM/3
21 CC663 000032 INTC/3
22 CC664 001135 FD/3
23 CC665 000757 CF/3
24 CC666 000431 DL/3
25 CC667 000332 DS/3
26 CC670 000443 AA/3
27 CC671 000462 SS/3
28 CC672 002447 DP/3
29 CC673 003010 MH/3
30 CC674 002316 LK/3
31 CC675 003044 IX/3
32 CC676 000032 INTC/3
33 CC677 000032 INTC/3
```

```
10016 .MAIN
01
02                .ENDC
03 CC700 000032 INTO/3
04 CC701 002274 DO/3
05 CC702 000250 EL/3
06 CC703 000227 HL/3
07 CC704 000073 LA/3
08 CC705 000100 LO/3
09 CC706 000105 LX/3
10 CC707 000112 WA/3
11 CC710 000121 WS/3
12 CC711 000053 AM/3
13 CC712 000130 WM/3
14 CC713 000057 AL/3
15 CC714 002472 XI/3
16 CC715 000631 JLL/3
17 CC716 000657 JDL/3
18 CC717 000702 JEL/3
19 CC720 000304 XL/3
20 CC721 000272 ES/3
21 CC722 000260 EA/3
22 CC723 000240 ZL/3
23 CC724 000066 RL/3
24 CC725 002245 SP/3
25 CC726 001351 RE/3
26 CC727 000316 RS/3
27 CC730 000165 WD/3
28 CC731 000412 RX/3
29 CC732 000360 HS/3
30 CC733 000376 XS/3
31 CC734 005225 GG/3
32 CC735 002257 DI/3
33 CC736 002403 AP/3
34 CC737 002367 UL/3
```

```
10017 .MAIN
01
02 CC740 000711 CI/3
03 CC741 000062 AC/3
04 CC742 000562 NS/3
05 CC743 000606 ND/3
06 CC744 000501 AS/3
07 CC745 000513 AD/3
08 CC746 000553 LS/3
09 CC747 000545 LD/3
10 CC750 002200 SH/3
11 CC751 002205 SL/3
12 CC752 002212 SE/3
13 CC753 002217 SN/3
14 CC754 002224 SO/3
15 CC755 002232 SZERO/3
16 CC756 002237 SX/3
17 CC757 005232 GP/3
18 CC760 001221 FA/3
19 CC761 001276 FS/3
20 CC762 001010 FM/3
21 CC763 000032 INTC/3
22 CC764 001135 FD/3
23 CC765 000757 CF/3
24 CC766 000431 DL/3
25 CC767 000352 DS/3
26 CC770 000443 AA/3
27 CC771 000462 SS/3
28 CC772 002447 DP/3
29 CC773 003010 MH/3
30 CC774 002316 LK/3
31 CC775 003044 IX/3
32 CC776 000032 INTC/3
33 CC777 000032 INTO/3
```

```
10018 .MAIN
01
02                .ENDC
03                                        ;ESCAPE MODE
04 01000 002303 FOOE/3                    ;FPU INSTALLED
05 01001 002272 DOE/3
06 01002 000246 ELE/3
07 01003 000225 HLE/3
08 01004 000071 LAE/3
09 01005 000076 LOE/3
10 01006 000103 LXE/3
11 01007 000110 WAE/3
12 01010 000117 WSE/3
13 01011 000051 AME/3
14 01012 000143 WMFPE/3
15 01013 000055 ALE/3
16 01014 002470 XIE/3
17 01015 000622 JLE/3
18 01016 000635 JDE/3
19 01017 000672 JEE/3
20 01020 000302 XLE/3
21 01021 000270 ESE/3
22 01022 000256 EAE/3
23 01023 000236 ZLE/3
24 01024 000064 RLE/3
25 01025 002243 SPE/3
26 01026 001347 REE/3
27 01027 000314 RSE/3
28 01030 000163 WDE/3
29 01031 000410 RXE/3
30 01032 000356 HSE/3
31 01033 000374 XSE/3
32 01034 005223 GGE/3
33 01035 002255 DIE/3
34 01036 002401 APE/3
35 01037 002365 ULE/3
```

```
10019 .MAIN
01
02 01040 000707 CIE/3
03 01041 000060 ACE/3
04 01042 000560 NSE/3
05 01043 000604 NDE/3
06 01044 000477 ASE/3
07 01045 000511 ADE/3
08 01046 000531 LSE/3
09 01047 000543 LDE/3
10 01050 002176 SHE/3
11 01051 002203 SLE/3
12 01052 002210 SEE/3
13 01053 002215 SNE/3
14 01054 002222 SOE/3
15 01055 002230 SZE/3
16 01056 002235 SXE/3
17 01057 005230 GPE/3
18 01060 001266 FAFPE/3
19 01061 001341 FSFPE/3
20 01062 001064 FMFPE/3
21 01063 002305 F63E/3
22 01064 001211 FDFPE/3
23 01065 000755 CFE/3
24 01066 000427 DLE/3
25 01067 000330 DSE/3
26 01070 000441 AAE/3
27 01071 000460 SSE/3
28 01072 002445 DPE/3
29 01073 003006 MHE/3
30 01074 002314 LKE/3
31 01075 003042 IXE/3
32 01076 002307 F76E/3
33 01077 002311 F77E/3
```

```
10020 .MAIN
01
02        000003 .DO 3              ; W-FIELD<>0
03 01100  002303 FCOE/3
04 01101  002272 DOE/3
05 01102  000246 ELE/3
06 01103  000225 HLE/3
07 01104  000071 LAE/3
08 01105  000076 LOE/3
09 01106  000103 LXE/3
10 01107  000110 WAE/3
11 01110  000117 WSE/3
12 01111  000051 AME/3
13 01112  000143 WMFPE/3
14 01113  000055 ALE/3
15 01114  002470 XIE/3
16 01115  000627 JLLE/3
17 01116  000655 JDLE/3
18 01117  000700 JELE/3
19 01120  000302 XLE/3
20 01121  000270 ESE/3
21 01122  000256 EAE/3
22 01123  000236 ZLE/3
23 01124  000064 RLE/3
24 01125  002243 SPE/3
25 01126  001347 REE/3
26 01127  000314 RSE/3
27 01130  000163 WDE/3
28 01131  000410 RXE/3
29 01132  000356 HSE/3
30 01133  000374 XSE/3
31 01134  005223 GGE/3
32 01135  002255 DIE/3
33 01136  002401 APE/3
34 01137  002365 ULE/3
```

```
10021 .MAIN
01
02 01140 000707 CIE/3
03 01141 000060 ACE/3
04 01142 000560 NSE/3
05 01143 000604 NDE/3
06 01144 000477 ASE/3
07 01145 000511 ADE/3
08 01146 000531 LSE/3
09 01147 000543 LDE/3
10 01150 002176 SHE/3
11 01151 002203 SLE/3
12 01152 002210 SEE/3
13 01153 002215 SNE/3
14 01154 002222 SOE/3
15 01155 002230 SZE/3
16 01156 002235 SXE/3
17 01157 005230 GPE/3
18 01160 001266 FAFPE/3
19 01161 001341 FSFPE/3
20 01162 001064 FMFPE/3
21 01163 002305 F63E/3
22 01164 001211 FDFPE/3
23 01165 000755 CFE/3
24 01166 000427 DLE/3
25 01167 000330 DSE/3
26 01170 000441 AAE/3
27 01171 000460 SSE/3
28 01172 002445 DPE/3
29 01173 003006 MHE/3
30 01174 002314 LKE/3
31 01175 003042 IXE/3
32 01176 002307 F76E/3
33 01177 002311 F77E/3
```

```
10022 .MAIN
01
02                .ENDC
03 01200 002303 FCGE/3
04 01201 002272 DOE/3
05 01202 000246 ELE/3
06 01203 000225 HLE/3
07 01204 000071 LAE/3
08 01205 000076 LOE/3
09 01206 000103 LXE/3
10 01207 000110 WAE/3
11 01210 000117 WSE/3
12 01211 000051 AME/3
13 01212 000143 WMFPE/3
14 01213 000055 ALE/3
15 01214 002470 XIE/3
16 01215 000627 JLLE/3
17 01216 000655 JDLE/3
18 01217 000700 JELE/3
19 01220 000302 XLE/3
20 01221 000270 ESE/3
21 01222 000256 EAE/3
22 01223 000236 ZLE/3
23 01224 000064 RLE/3
24 01225 002243 SPE/3
25 01226 001347 REE/3
26 01227 000314 RSE/3
27 01230 000163 WDE/3
28 01231 000410 RXE/3
29 01232 000356 HSE/3
30 01233 000374 XSE/3
31 01234 005223 GGE/3
32 01235 002255 DIE/3
33 01236 002401 APE/3
34 01237 002365 ULE/3
```

```
10023 .MAIN
01
02 01240 000707 CIE/3
03 01241 000060 ACE/3
04 01242 000560 NSE/3
05 01243 000604 NDE/3
06 01244 000477 ASE/3
07 01245 000511 ADE/3
08 01246 000531 LSE/3
09 01247 000543 LDE/3
10 01250 002176 SHE/3
11 01251 002203 SLE/3
12 01252 002210 SEE/3
13 01253 002215 SNE/3
14 01254 002222 SOE/3
15 01255 002230 SZE/3
16 01256 002235 SXE/3
17 01257 005230 GPE/3
18 01260 001266 FAFPE/3
19 01261 001341 FSFPE/3
20 01262 001064 FMFPE/3
21 01263 002305 F63E/3
22 01264 001211 FDFPE/3
23 01265 000755 CFE/3
24 01266 000427 DLE/3
25 01267 000330 DSE/3
26 01270 000441 AAE/3
27 01271 000460 SSE/3
28 01272 002445 DPE/3
29 01273 003006 MHE/3
30 01274 002314 LKE/3
31 01275 003042 IXE/3
32 01276 002307 F76E/3
33 01277 002311 F77E/3
```

```
10024 .MAIN
01
02                .ENDC
03 01300 002303 FCOE/3
04 01301 002272 DOE/3
05 01302 000246 ELE/3
06 01303 000225 HLE/3
07 01304 000071 LAE/3
08 01305 000076 LOE/3
09 01306 000103 LXE/3
10 01307 000110 WAE/3
11 01310 000117 WSE/3
12 01311 000051 AME/3
13 01312 000143 WMFPE/3
14 01313 000055 ALE/3
15 01314 002470 XIE/3
16 01315 000627 JLLE/3
17 01316 000655 JDLE/3
18 01317 000700 JELE/3
19 01320 000302 XLE/3
20 01321 000270 ESE/3
21 01322 000256 EAE/3
22 01323 000236 ZLE/3
23 01324 000064 RLE/3
24 01325 002243 SPE/3
25 01326 001347 REE/3
26 01327 000314 RSE/3
27 01330 000163 WDE/3
28 01331 000410 RXE/3
29 01332 000356 HSE/3
30 01333 000374 XSE/3
31 01334 005223 GGE/3
32 01335 002255 DIE/3
33 01336 002401 APE/3
34 01337 002365 ULE/3
```

```
10025 .MAIN
01
02 01340 000707 CIE/3
03 01341 000060 ACE/3
04 01342 000560 NSE/3
05 01343 000604 NDE/3
06 01344 000477 ASE/3
07 01345 000511 ADE/3
08 01346 000531 LSE/3
09 01347 000543 LDE/3
10 01350 002176 SHE/3
11 01351 002203 SLE/3
12 01352 002210 SEE/3
13 01353 002215 SNE/3
14 01354 002222 SOE/3
15 01355 002230 SZE/3
16 01356 002235 SXE/3
17 01357 005230 GPE/3
18 01360 001266 FAFPE/3
19 01361 001341 FSFPE/3
20 01362 001064 FMFPE/3
21 01363 002305 F63E/3
22 01364 001211 FDFPE/3
23 01365 000755 CFE/3
24 01366 000427 DLE/3
25 01367 000330 DSE/3
26 01370 000441 AAE/3
27 01371 000460 SSE/3
28 01372 002445 DPE/3
29 01373 003006 MHE/3
30 01374 002314 LKE/3
31 01375 003042 IXE/3
32 01376 002307 F76E/3
33 01377 002311 F77E/3
```

```
10026 .MAIN
01
02                .ENDC
03 01400 002303 FUUE/3          ;FPU NOT INSTALLED
04 01401 002272 DOE/3
05 01402 000246 ELE/3
06 01403 000225 HLE/3
07 01404 000071 LAE/3
08 01405 000076 LOE/3
09 01406 000103 LXE/3
10 01407 000110 WAE/3
11 01410 000117 WSE/3
12 01411 000051 AME/3
13 01412 000126 WME/3
14 01413 000055 ALE/3
15 01414 002470 XIE/3
16 01415 000622 JLE/3
17 01416 000635 JDE/3
18 01417 000672 JEE/3
19 01420 000302 XLE/3
20 01421 000270 ESE/3
21 01422 000256 EAE/3
22 01423 000236 ZLE/3
23 01424 000064 RLE/3
24 01425 002243 SPE/3
25 01426 001347 REE/3
26 01427 000314 RSE/3
27 01430 000163 WDE/3
28 01431 000410 RXE/3
29 01432 000356 HSE/3
30 01433 000374 XSE/3
31 01434 005223 GGE/3
32 01435 002255 DIE/3
33 01436 002401 APE/3
34 01437 002365 ULE/3
```

```
10027 .MAIN
01
02 01440 000707 CIE/3
03 01441 000060 ACE/3
04 01442 000560 NSE/3
05 01443 000604 NDE/3
06 01444 000477 ASE/3
07 01445 000511 ADE/3
08 01446 000531 LSE/3
09 01447 000543 LDE/3
10 01450 002176 SHE/3
11 01451 002203 SLE/3
12 01452 002210 SEE/3
13 01453 002215 SNE/3
14 01454 002222 SOE/3
15 01455 002230 SZE/3
16 01456 002235 SXE/3
17 01457 005230 GPE/3
18 01460 001217 FAE/3
19 01461 001274 FSE/3
20 01462 001006 FME/3
21 01463 002305 F63E/3
22 01464 001133 FDE/3
23 01465 000755 CFE/3
24 01466 000427 DLE/3
25 01467 000330 DSE/3
26 01470 000441 AAE/3
27 01471 000460 SSE/3
28 01472 002445 DPE/3
29 01473 003006 MHE/3
30 01474 002314 LKE/3
31 01475 003042 IXE/3
32 01476 002307 F76E/3
33 01477 002311 F77E/3
```

```
10028 .MAIN
01
02        000003 .DO 3                ; A=FIELD<>0
03 01500  002303 FCCE/3
04 01501  002272 DOE/3
05 01502  000246 ELE/3
06 01503  000225 HLE/3
07 01504  000071 LAE/3
08 01505  000076 LOE/3
09 01506  000103 LXE/3
10 01507  000110 WAE/3
11 01510  000117 WSE/3
12 01511  000051 AME/3
13 01512  000126 WME/3
14 01513  000055 ALE/3
15 01514  002470 XIE/3
16 01515  000627 JLLE/3
17 01516  000655 JDLE/3
18 01517  000700 JELE/3
19 01520  000302 XLE/3
20 01521  000270 ESE/3
21 01522  000256 EAE/3
22 01523  000236 ZLE/3
23 01524  000064 RLE/3
24 01525  002243 SPE/3
25 01526  001347 REE/3
26 01527  000314 RSE/3
27 01530  000163 WDE/3
28 01531  000410 RXE/3
29 01532  000356 HSE/3
30 01533  000374 XSE/3
31 01534  005223 GGE/3
32 01535  002255 DIE/3
33 01536  002401 APE/3
34 01537  002365 ULE/3
```

```
10029 .MAIN
01
02 01540 000707 CIE/3
03 01541 000060 ACE/3
04 01542 000560 NSE/3
05 01543 000604 NDE/3
06 01544 000477 ASE/3
07 01545 000511 ADE/3
08 01546 000531 LSE/3
09 01547 000543 LDE/3
10 01550 002176 SHE/3
11 01551 002203 SLE/3
12 01552 002210 SEE/3
13 01553 002215 SNE/3
14 01554 002222 SOE/3
15 01555 002230 SZE/3
16 01556 002235 SXE/3
17 01557 005230 GPE/3
18 01560 001217 FAE/3
19 01561 001274 FSE/3
20 01562 001006 FME/3
21 01563 002305 F63E/3
22 01564 001133 FDE/3
23 01565 000755 CFE/3
24 01566 000427 DLE/3
25 01567 000330 DSE/3
26 01570 000441 AAE/3
27 01571 000460 SSE/3
28 01572 002445 DPE/3
29 01573 003006 MHE/3
30 01574 002314 LKE/3
31 01575 003042 IXE/3
32 01576 002307 F76E/3
33 01577 002311 F77E/3
```

```
10030 .MAIN
01
02                .ENDC
03 01600 002303 FOOE/3
04 01601 002272 DOE/3
05 01602 000246 ELE/3
06 01603 000225 HLE/3
07 01604 000071 LAE/3
08 01605 000076 LOE/3
09 01606 000103 LXE/3
10 01607 000110 WAE/3
11 01610 000117 WSE/3
12 01611 000051 AME/3
13 01612 000126 WME/3
14 01613 000055 ALE/3
15 01614 002470 XIE/3
16 01615 000627 JLLE/3
17 01616 000655 JDLE/3
18 01617 000700 JELE/3
19 01620 000302 XLE/3
20 01621 000270 ESE/3
21 01622 000256 EAE/3
22 01623 000236 ZLE/3
23 01624 000064 RLE/3
24 01625 002243 SPE/3
25 01626 001347 REE/3
26 01627 000314 RSE/3
27 01630 000163 WDE/3
28 01631 000410 RXE/3
29 01632 000356 HSE/3
30 01633 000374 XSE/3
31 01634 005223 GGE/3
32 01635 002255 DIE/3
33 01636 002401 APE/3
34 01637 002365 ULE/3
```

```
10031 .MAIN
01
02 01640 000707 CIE/3
03 01641 000060 ACE/3
04 01642 000560 NSE/3
05 01643 000604 NDE/3
06 01644 000477 ASE/3
07 01645 000511 ADE/3
08 01646 000551 LSE/3
09 01647 000543 LDE/3
10 01650 002176 SHE/3
11 01651 002203 SLE/3
12 01652 002210 SEE/3
13 01653 002215 SNE/3
14 01654 002222 SOE/3
15 01655 002230 SZE/3
16 01656 002235 SXE/3
17 01657 005230 GPE/3
18 01660 001217 FAE/3
19 01661 001274 FSE/3
20 01662 001006 FME/3
21 01663 002305 F63E/3
22 01664 001133 FDE/3
23 01665 000755 CFE/3
24 01666 000427 DLE/3
25 01667 000330 DSE/3
26 01670 000441 AAE/3
27 01671 000460 SSE/3
28 01672 002445 DPE/3
29 01673 003006 MHE/3
30 01674 002314 LKE/3
31 01675 003042 IXE/3
32 01676 002307 F76E/3
33 01677 002311 F77E/3
```

```
10032 .MAIN
01
02                .ENDC
03 01700 002303 FOOE/3
04 01701 002272 DOE/3
05 01702 000246 ELE/3
06 01703 000225 HLE/3
07 01704 000071 LAE/3
08 01705 000076 LOE/3
09 01706 000103 LXE/3
10 01707 000110 WAE/3
11 01710 000117 WSE/3
12 01711 000051 AME/3
13 01712 000126 WME/3
14 01713 000055 ALE/3
15 01714 002470 XIE/3
16 01715 000627 JLLE/3
17 01716 000655 JDLE/3
18 01717 000700 JELE/3
19 01720 000302 XLE/3
20 01721 000270 ESE/3
21 01722 000256 EAE/3
22 01723 000236 ZLE/3
23 01724 000064 RLE/3
24 01725 002243 SPE/3
25 01726 001347 REE/3
26 01727 000314 RSE/3
27 01730 000163 WDE/3
28 01731 000410 RXE/3
29 01732 000356 HSE/3
30 01733 000374 XSE/3
31 01734 005223 GGE/3
32 01735 002255 DIE/3
33 01736 002401 APE/3
34 01737 002365 ULE/3
```

```
10033 .MAIN
01
02 01740 000707 CIE/3
03 01741 000060 ACE/3
04 01742 000560 NSE/3
05 01743 000604 NDE/3
06 01744 000477 ASE/3
07 01745 000511 ADE/3
08 01746 000531 LSE/3
09 01747 000543 LDE/3
10 01750 002176 SHE/3
11 01751 002203 SLE/3
12 01752 002210 SEE/3
13 01753 002215 SNE/3
14 01754 002222 SOE/3
15 01755 002230 SZE/3
16 01756 002235 SXE/3
17 01757 005230 GPE/3
18 01760 001217 FAE/3
19 01761 001274 FSE/3
20 01762 001006 FME/3
21 01763 002305 F63E/3
22 01764 001133 FDE/3
23 01765 000755 CFE/3
24 01766 000427 DLE/3
25 01767 000330 DSE/3
26 01770 000441 AAE/3
27 01771 000460 SSE/3
28 01772 002445 DPE/3
29 01773 003006 MHE/3
30 01774 002314 LKE/3
31 01775 003042 IXE/3
32 01776 002307 F76E/3
33 01777 002311 F77E/3
```

```
10034 .MAIN
01
02                .ENDC
03                .END

0000 SOURCE LINES IN ERROR
```

```
 0001 .MAIN DOMUS MACRO ASSEMBLER REV 02.00
01
02                ;ADDRESS CALCULATION DECODING MAP
03                ;
04                ; 0: PREFETCH ERROR
05                ; 1: ZERO
06                ; 2: ESCAPE MODE
07                ; 3: AFTER ESCAPE
08                ; 4: AFTER AM
09                ; 5: INSTR BIT (8)
10                ; 6:           (9)
11                ; 7:           (10)
12                ; 8:           (11)
13
```

```
10002 .MAIN
01        000000 .LOC 0
02
03        000002 .DO 2
04 CC000 002013 DIR/3
05 CC001 002015 INDEX/3
06 CC002 002015 INDEX/3
07 CC003 002015 INDEX/3
08 CC004 002017 DIRI/3
09 CC005 002040 INDXI/3
10 CC006 002040 INDXI/3
11 CC007 002040 INDXI/3
12
13 CC010 002014 REL/3
14 CC011 002016 RELX/3
15 CC012 002016 RELX/3
16 CC013 002016 RELX/3
17 CC014 002032 RELI/3
18 CC015 002046 RELXI/3
19 CC016 002046 RELXI/3
20 CC017 002046 RELXI/3
21
22 CC020 002112 AMD/3
23 CC021 002117 AMX/3
24 CC022 002117 AMX/3
25 CC023 002117 AMX/3
26 CC024 002127 AMDI/3
27 CC025 002142 AMXI/3
28 CC026 002142 AMXI/3
29 CC027 002142 AMXI/3
30
31 CC030 002110 AMR/3
32 CC031 002114 AMRX/3
33 CC032 002114 AMRX/3
34 CC033 002114 AMRX/3
35 CC034 002122 AMRI/3
36 CC035 002134 AMRXI/3
37 CC036 002134 AMRXI/3
38 CC037 002134 AMRXI/3
```

```
10003 .MAIN
01       000020 .DO 16.          ; AFTER ESCAPE=1 ESCAPE=0
02 00040 002003 AENES/3
03 00041 002003 AENES/3
04 00042 002003 AENES/3
05 00043 002003 AENES/3
06 00044 002003 AENES/3
07 00045 002003 AENES/3
08 00046 002003 AENES/3
09 00047 002003 AENES/3
10 00050 002003 AENES/3
11 00051 002003 AENES/3
12 00052 002003 AENES/3
13 00053 002003 AENES/3
14 00054 002003 AENES/3
15 00055 002003 AENES/3
16 00056 002003 AENES/3
17 00057 002003 AENES/3
18       000020 .DO 16.          ; AFTER ESCAPE=1 ESCAPE=0 AFTER AM=1
19 00060 002005 INESC/3
20 00061 002005 INESC/3
21 00062 002005 INESC/3
22 00063 002005 INESC/3
23 00064 002005 INESC/3
24 00065 002005 INESC/3
25 00066 002005 INESC/3
26 00067 002005 INESC/3
27 00070 002005 INESC/3
28 00071 002005 INESC/3
29 00072 002005 INESC/3
30 00073 002005 INESC/3
31 00074 002005 INESC/3
32 00075 002005 INESC/3
33 00076 002005 INESC/3
34 00077 002005 INESC/3
```

```
10004 .MAIN
01
02 CC100 002013 DIR/3              ;ESCAPE = 1 AFTER ESCAPE =0
03 CC101 002015 INDEX/3
04 CC102 002015 INDEX/3
05 CC103 002015 INDEX/3
06 CC104 002054 DIRIE/3
07 CC105 002072 IDXIE/3
08 CC106 002072 IDXIE/3
09 CC107 002072 IDXIE/3
10
11 CC110 002014 REL/3
12 CC111 002016 RELX/3
13 CC112 002016 RELX/3
14 CC113 002016 RELX/3
15 CC114 002063 RELIE/3
16 CC115 002101 RLXIE/3
17 CC116 002101 RLXIE/3
18 CC117 002101 RLXIE/3
19
20 CC120 002112 AMD/3
21 CC121 002117 AMX/3
22 CC122 002117 AMX/3
23 CC123 002117 AMX/3
24 CC124 002151 AMDIE/3
25 CC125 002160 AMXIE/3
26 CC126 002160 AMXIE/3
27 CC127 002160 AMXIE/3
28
29 CC130 002110 AMR/3
30 CC131 002114 AMRX/3
31 CC132 002114 AMRX/3
32 CC133 002114 AMRX/3
33 CC134 002150 AMRIE/3
34 CC135 002157 ARXIE/3
35 CC136 002157 ARXIE/3
36 CC137 002157 ARXIE/3
```

```
10005 .MAIN
01
02        000020 .DO 16.          ; AFTER ESCAPE=1 ESCAPE=1
03 00140 002002 AFTES/3
04 00141 002002 AFTES/3
05 00142 002002 AFTES/3
06 00143 002002 AFTES/3
07 00144 002002 AFTES/3
08 00145 002002 AFTES/3
09 00146 002002 AFTES/3
10 00147 002002 AFTES/3
11 00150 002002 AFTES/3
12 00151 002002 AFTES/3
13 00152 002002 AFTES/3
14 00153 002002 AFTES/3
15 00154 002002 AFTES/3
16 00155 002002 AFTES/3
17 00156 002002 AFTES/3
18 00157 002002 AFTES/3
19        000020 .DO 16.          ; AFTER ESCAPE=1 ESCAPE=1 AFTER AM=1
20 00160 002005 INESC/3
21 00161 002005 INESC/3
22 00162 002005 INESC/3
23 00163 002005 INESC/3
24 00164 002005 INESC/3
25 00165 002005 INESC/3
26 00166 002005 INESC/3
27 00167 002005 INESC/3
28 00170 002005 INESC/3
29 00171 002005 INESC/3
30 00172 002005 INESC/3
31 00173 002005 INESC/3
32 00174 002005 INESC/3
33 00175 002005 INESC/3
34 00176 002005 INESC/3
35 00177 002005 INESC/3
```

```
100C6 .MAIN
01
02                .ENDC
03 CC200 002013 DIR/3
04 CC201 002015 INDEX/3
05 CC202 002015 INDEX/3
06 CC203 002015 INDEX/3
07 CC204 002017 DIRI/3
08 CC205 002040 INDXI/3
09 CC206 002040 INDXI/3
10 CC207 002040 INDXI/3
11
12 CC210 002014 REL/3
13 CC211 002016 RELX/3
14 CC212 002016 RELX/3
15 CC213 002016 RELX/3
16 CC214 002032 RELI/3
17 CC215 002046 RELXI/3
18 CC216 002046 RELXI/3
19 CC217 002046 RELXI/3
20
21 CC220 002112 AMD/3
22 CC221 002117 AMX/3
23 CC222 002117 AMX/3
24 CC223 002117 AMX/3
25 CC224 002127 AMDI/3
26 CC225 002142 AMXI/3
27 CC226 002142 AMXI/3
28 CC227 002142 AMXI/3
29
30 CC230 002110 AMR/3
31 CC231 002114 AMRX/3
32 CC232 002114 AMRX/3
33 CC233 002114 AMRX/3
34 CC234 002122 AMRI/3
35 CC235 002134 AMRXI/3
36 CC236 002134 AMRXI/3
37 CC237 002134 AMRXI/3
```

```
10CC7 .MAIN
01        000020 .DO 16.          ; AFTER ESCAPE=1 ESCAPE=0
02 CC240 002003 AENES/3
03 CC241 002003 AENES/3
04 CC242 002003 AENES/3
05 CC243 002003 AENES/3
06 CC244 002003 AENES/3
07 CC245 002003 AENES/3
08 CC246 002003 AENES/3
09 CC247 002003 AENES/3
10 CC250 002003 AENES/3
11 CC251 002003 AENES/3
12 CC252 002003 AENES/3
13 CC253 002003 AENES/3
14 CC254 002003 AENES/3
15 CC255 002003 AENES/3
16 CC256 002003 AENES/3
17 CC257 002003 AENES/3
18        000020 .DO 16.          ; AFTER ESCAPE=1 ESCAPE=0 AFTER AM=1
19 CC260 002005 INESC/3
20 CC261 002005 INESC/3
21 CC262 002005 INESC/3
22 CC263 002005 INESC/3
23 CC264 002005 INESC/3
24 CC265 002005 INESC/3
25 CC266 002005 INESC/3
26 CC267 002005 INESC/3
27 CC270 002005 INESC/3
28 CC271 002005 INESC/3
29 CC272 002005 INESC/3
30 CC273 002005 INESC/3
31 CC274 002005 INESC/3
32 CC275 002005 INESC/3
33 CC276 002005 INESC/3
34 CC277 002005 INESC/3
```

```
!00C8 .MAIN
01
02 C0300 002013 DIR/3              ;ESCAPE = 1 AFTER ESCAPE =0
03 C0301 002015 INDEX/3
04 C0302 002015 INDEX/3
05 C0303 002015 INDEX/3
06 C0304 002054 DIRIE/3
07 C0305 002072 IDXIE/3
08 C0306 002072 IDXIE/3
09 C0307 002072 IDXIE/3
10
11 C0310 002014 REL/3
12 C0311 002016 RELX/3
13 C0312 002016 RELX/3
14 C0313 002016 RELX/3
15 C0314 002063 RELIE/3
16 C0315 002101 RLXIE/3
17 C0316 002101 RLXIE/3
18 C0317 002101 RLXIE/3
19
20 C0320 002112 AMD/3
21 C0321 002117 AMX/3
22 C0322 002117 AMX/3
23 C0323 002117 AMX/3
24 C0324 002151 AMDIE/3
25 C0325 002160 AMXIE/3
26 C0326 002160 AMXIE/3
27 C0327 002160 AMXIE/3
28
29 C0330 002110 AMR/3
30 C0331 002114 AMRX/3
31 C0332 002114 AMRX/3
32 C0333 002114 AMRX/3
33 C0334 002150 AMRIE/3
34 C0335 002157 ARXIE/3
35 C0336 002157 ARXIE/3
36 C0337 002157 ARXIE/3
```

```
10009 .MAIN
01
02        000020 .DO 16.          ; AFTER ESCAPE=1 ESCAPE=1
03 C0340  002002 AFTES/3
04 C0341  002002 AFTES/3
05 C0342  002002 AFTES/3
06 C0343  002002 AFTES/3
07 C0344  002002 AFTES/3
08 C0345  002002 AFTES/3
09 C0346  002002 AFTES/3
10 C0347  002002 AFTES/3
11 C0350  002002 AFTES/3
12 C0351  002002 AFTES/3
13 C0352  002002 AFTES/3
14 C0353  002002 AFTES/3
15 C0354  002002 AFTES/3
16 C0355  002002 AFTES/3
17 C0356  002002 AFTES/3
18 C0357  002002 AFTES/3
19        000020 .DO 16.          ; AFTER ESCAPE=1 ESCAPE=1 AFTER AM=1
20 C0360  002005 INESC/3
21 C0361  002005 INESC/3
22 C0362  002005 INESC/3
23 C0363  002005 INESC/3
24 C0364  002005 INESC/3
25 C0365  002005 INESC/3
26 C0366  002005 INESC/3
27 C0367  002005 INESC/3
28 C0370  002005 INESC/3
29 C0371  002005 INESC/3
30 C0372  002005 INESC/3
31 C0373  002005 INESC/3
32 C0374  002005 INESC/3
33 C0375  002005 INESC/3
34 C0376  002005 INESC/3
35 C0377  002005 INESC/3
```

```
10010 .MAIN
01
02                .ENDC
03        000010 .DO 8.           ; PREFETCH ERROR
04        000040 .DO 32.
05 CC400 000036 PFINT/3
06 CC401 000036 PFINT/3
07 CC402 000036 PFINT/3
08 CC403 000036 PFINT/3
09 CC404 000036 PFINT/3
10 CC405 000036 PFINT/3
11 CC406 000036 PFINT/3
12 CC407 000036 PFINT/3
13 CC410 000036 PFINT/3
14 CC411 000036 PFINT/3
15 CC412 000036 PFINT/3
16 CC413 000036 PFINT/3
17 CC414 000036 PFINT/3
18 CC415 000036 PFINT/3
19 CC416 000036 PFINT/3
20 CC417 000036 PFINT/3
21 CC420 000036 PFINT/3
22 CC421 000036 PFINT/3
23 CC422 000036 PFINT/3
24 CC423 000036 PFINT/3
25 CC424 000036 PFINT/3
26 CC425 000036 PFINT/3
27 CC426 000036 PFINT/3
28 CC427 000036 PFINT/3
29 CC430 000036 PFINT/3
30 CC431 000036 PFINT/3
31 CC432 000036 PFINT/3
32 CC433 000036 PFINT/3
33 CC434 000036 PFINT/3
34 CC435 000036 PFINT/3
35 CC436 000036 PFINT/3
36 CC437 000036 PFINT/3
```

```
10011 .MAIN
01
02                .ENDC
03        000040  .DO 32.
04 CC440  000036  PFINT/3
05 CC441  000036  PFINT/3
06 CC442  000036  PFINT/3
07 CC443  000036  PFINT/3
08 CC444  000036  PFINT/3
09 CC445  000036  PFINT/3
10 CC446  000036  PFINT/3
11 CC447  000036  PFINT/3
12 CC450  000036  PFINT/3
13 CC451  000036  PFINT/3
14 CC452  000036  PFINT/3
15 CC453  000036  PFINT/3
16 CC454  000036  PFINT/3
17 CC455  000036  PFINT/3
18 CC456  000036  PFINT/3
19 CC457  000036  PFINT/3
20 CC460  000036  PFINT/3
21 CC461  000036  PFINT/3
22 CC462  000036  PFINT/3
23 CC463  000036  PFINT/3
24 CC464  000036  PFINT/3
25 CC465  000036  PFINT/3
26 CC466  000036  PFINT/3
27 CC467  000036  PFINT/3
28 CC470  000036  PFINT/3
29 CC471  000036  PFINT/3
30 CC472  000036  PFINT/3
31 CC473  000036  PFINT/3
32 CC474  000036  PFINT/3
33 CC475  000036  PFINT/3
34 CC476  000036  PFINT/3
35 CC477  000036  PFINT/3
```

```
!0012 .MAIN
01
02                .ENDC
03        000040 .DO 32.
04 CC500 000036 PFINT/3
05 CC501 000036 PFINT/3
06 CC502 000036 PFINT/3
07 CC503 000036 PFINT/3
08 CC504 000036 PFINT/3
09 CC505 000036 PFINT/3
10 CC506 000036 PFINT/3
11 CC507 000036 PFINT/3
12 CC510 000036 PFINT/3
13 CC511 000036 PFINT/3
14 CC512 000036 PFINT/3
15 CC513 000036 PFINT/3
16 CC514 000036 PFINT/3
17 CC515 000036 PFINT/3
18 CC516 000036 PFINT/3
19 CC517 000036 PFINT/3
20 CC520 000036 PFINT/3
21 CC521 000036 PFINT/3
22 CC522 000036 PFINT/3
23 CC523 000036 PFINT/3
24 CC524 000036 PFINT/3
25 CC525 000036 PFINT/3
26 CC526 000036 PFINT/3
27 CC527 000036 PFINT/3
28 CC530 000036 PFINT/3
29 CC531 000036 PFINT/3
30 CC532 000036 PFINT/3
31 CC533 000036 PFINT/3
32 CC534 000036 PFINT/3
33 CC535 000036 PFINT/3
34 CC536 000036 PFINT/3
35 CC537 000036 PFINT/3
```

```
10013 .MAIN
01
02                .ENDC
03        000040  .DO 32.
04  CC540 000036  PFINT/3
05  CC541 000036  PFINT/3
06  CC542 000036  PFINT/3
07  CC543 000036  PFINT/3
08  CC544 000036  PFINT/3
09  CC545 000036  PFINT/3
10  CC546 000036  PFINT/3
11  CC547 000036  PFINT/3
12  CC550 000036  PFINT/3
13  CC551 000036  PFINT/3
14  CC552 000036  PFINT/3
15  CC553 000036  PFINT/3
16  CC554 000036  PFINT/3
17  CC555 000036  PFINT/3
18  CC556 000036  PFINT/3
19  CC557 000036  PFINT/3
20  CC560 000036  PFINT/3
21  CC561 000036  PFINT/3
22  CC562 000036  PFINT/3
23  CC563 000036  PFINT/3
24  CC564 000036  PFINT/3
25  CC565 000036  PFINT/3
26  CC566 000036  PFINT/3
27  CC567 000036  PFINT/3
28  CC570 000036  PFINT/3
29  CC571 000036  PFINT/3
30  CC572 000036  PFINT/3
31  CC573 000036  PFINT/3
32  CC574 000036  PFINT/3
33  CC575 000036  PFINT/3
34  CC576 000036  PFINT/3
35  CC577 000036  PFINT/3
```

```
10014 .MAIN
01
02                .ENDC
03        000040  .DO 32.
04  CC600 000036  PFINT/3
05  CC601 000036  PFINT/3
06  CC602 000036  PFINT/3
07  CC603 000036  PFINT/3
08  CC604 000036  PFINT/3
09  CC605 000036  PFINT/3
10  CC606 000036  PFINT/3
11  CC607 000036  PFINT/3
12  CC610 000036  PFINT/3
13  CC611 000036  PFINT/3
14  CC612 000036  PFINT/3
15  CC613 000036  PFINT/3
16  CC614 000036  PFINT/3
17  CC615 000036  PFINT/3
18  CC616 000036  PFINT/3
19  CC617 000036  PFINT/3
20  CC620 000036  PFINT/3
21  CC621 000036  PFINT/3
22  CC622 000036  PFINT/3
23  CC623 000036  PFINT/3
24  CC624 000036  PFINT/3
25  CC625 000036  PFINT/3
26  CC626 000036  PFINT/3
27  CC627 000036  PFINT/3
28  CC630 000036  PFINT/3
29  CC631 000036  PFINT/3
30  CC632 000036  PFINT/3
31  CC633 000036  PFINT/3
32  CC634 000036  PFINT/3
33  CC635 000036  PFINT/3
34  CC636 000036  PFINT/3
35  CC637 000036  PFINT/3
```

```
10015 .MAIN
01
02                .ENDC
03         000040 .DO 32.
04 CC640 000036 PFINT/3
05 CC641 000036 PFINT/3
06 CC642 000036 PFINT/3
07 CC643 000036 PFINT/3
08 CC644 000036 PFINT/3
09 CC645 000036 PFINT/3
10 CC646 000036 PFINT/3
11 CC647 000036 PFINT/3
12 CC650 000036 PFINT/3
13 CC651 000036 PFINT/3
14 CC652 000036 PFINT/3
15 CC653 000036 PFINT/3
16 CC654 000036 PFINT/3
17 CC655 000036 PFINT/3
18 CC656 000036 PFINT/3
19 CC657 000036 PFINT/3
20 CC660 000036 PFINT/3
21 CC661 000036 PFINT/3
22 CC662 000036 PFINT/3
23 CC663 000036 PFINT/3
24 CC664 000036 PFINT/3
25 CC665 000036 PFINT/3
26 CC666 000036 PFINT/3
27 CC667 000036 PFINT/3
28 CC670 000036 PFINT/3
29 CC671 000036 PFINT/3
30 CC672 000036 PFINT/3
31 CC673 000036 PFINT/3
32 CC674 000036 PFINT/3
33 CC675 000036 PFINT/3
34 CC676 000036 PFINT/3
35 CC677 000036 PFINT/3
```

```
10016 .MAIN
01
02                .ENDC
03        000040 .DO 32.
04 CC700  000036 PFINT/3
05 CC701  000036 PFINT/3
06 CC702  000036 PFINT/3
07 CC703  000036 PFINT/3
08 CC704  000036 PFINT/3
09 CC705  000036 PFINT/3
10 CC706  000036 PFINT/3
11 CC707  000036 PFINT/3
12 CC710  000036 PFINT/3
13 CC711  000036 PFINT/3
14 CC712  000036 PFINT/3
15 CC713  000036 PFINT/3
16 CC714  000036 PFINT/3
17 CC715  000036 PFINT/3
18 CC716  000036 PFINT/3
19 CC717  000036 PFINT/3
20 CC720  000036 PFINT/3
21 CC721  000036 PFINT/3
22 CC722  000036 PFINT/3
23 CC723  000036 PFINT/3
24 CC724  000036 PFINT/3
25 CC725  000036 PFINT/3
26 CC726  000036 PFINT/3
27 CC727  000036 PFINT/3
28 CC730  000036 PFINT/3
29 CC731  000036 PFINT/3
30 CC732  000036 PFINT/3
31 CC733  000036 PFINT/3
32 CC734  000036 PFINT/3
33 CC735  000036 PFINT/3
34 CC736  000036 PFINT/3
35 CC737  000036 PFINT/3
```

```
!0017 .MAIN
01
02                .ENDC
03        000040 .DO 32.
04 CC740  000036 PFINT/3
05 CC741  000036 PFINT/3
06 CC742  000036 PFINT/3
07 CC743  000036 PFINT/3
08 CC744  000036 PFINT/3
09 CC745  000036 PFINT/3
10 CC746  000036 PFINT/3
11 CC747  000036 PFINT/3
12 CC750  000036 PFINT/3
13 CC751  000036 PFINT/3
14 CC752  000036 PFINT/3
15 CC753  000036 PFINT/3
16 CC754  000036 PFINT/3
17 CC755  000036 PFINT/3
18 CC756  000036 PFINT/3
19 CC757  000036 PFINT/3
20 CC760  000036 PFINT/3
21 CC761  000036 PFINT/3
22 CC762  000036 PFINT/3
23 CC763  000036 PFINT/3
24 CC764  000036 PFINT/3
25 CC765  000036 PFINT/3
26 CC766  000036 PFINT/3
27 CC767  000036 PFINT/3
28 CC770  000036 PFINT/3
29 CC771  000036 PFINT/3
30 CC772  000036 PFINT/3
31 CC773  000036 PFINT/3
32 CC774  000036 PFINT/3
33 CC775  000036 PFINT/3
34 CC776  000036 PFINT/3
35 CC777  000036 PFINT/3
```

```
10018 .MAIN
01
02                .ENDC
03                .END

0000 SOURCE LINES IN ERROR
```

```
0001 .MAIN DOMUS MACRO ASSEMBLER REV 02.00
01
02                ;CPU823 PREFETCH AND OPERANDFETCH DECODING MAP
03
04                ; ADDRESS
05
06                ; 0: PREFETCH ERROR
07                ; 1: 0==UNCONDITIONAL PREFETCH
08                ; 2: (ESCAPE)
09                ; 3: INSTRUCTION C
10                ; 4:               .1
11                ; 5:                2
12                ; 6:                3
13                ; 7:               .4
14                ; 8:               .5
15
16                ; OUTPUT
17
18                ; 0: NOT USED
19                ; 1: ENABLE PREFETCH
20                ; 2: ENABLE OPERANDFETCH
21                ; 3: JUMP
22
23          000004 PREFETCH=1B13.
24          000002 READOP=1B14.
25          000001 JUMP=1B15.
```

```
10002 .MAIN
01
02        000002 .DO 2                          ; UNCONDITIONAL PREFETCH
03 CC000  000004 PREFETCH                       ; UNASSIGNED
04 CC001  000004 PREFETCH                       ; DO
05 CC002  000006 PREFETCH+READOP                ; EL
06 CC003  000006 PREFETCH+READOP                ; HL
07 CC004  000006 PREFETCH+READOP                ; LA
08 CC005  000006 PREFETCH+READOP                ; LO
09 CC006  000006 PREFETCH+READOP                ; LX
10 CC007  000006 PREFETCH+READOP                ; WA
11 CC010  000006 PREFETCH+READOP                ; WS
12 CC011  000004 PREFETCH                       ; AM
13 CC012  000006 PREFETCH+READOP                ; WM
14 CC013  000004 PREFETCH                       ; AL
15 CC014  000004 PREFETCH                       ; XI
16 CC015  000007 PREFETCH+READOP+JUMP           ; JL
17 CC016  000007 PREFETCH+READOP+JUMP           ; JD
18 CC017  000007 PREFETCH+READOP+JUMP           ; JE
19 CC020  000006 PREFETCH+READOP                ; XL
20 CC021  000006 PREFETCH+READOP                ; ES
21 CC022  000006 PREFETCH+READOP                ; EA
22 CC023  000006 PREFETCH+READOP                ; ZL
23 CC024  000006 PREFETCH+READOP                ; RL
24 CC025  000004 PREFETCH                       ; SP
25 CC026  000006 PREFETCH+READOP                ; RE
26 CC027  000004 PREFETCH                       ; RS
27 CC030  000006 PREFETCH+READOP                ; WD
28 CC031  000006 PREFETCH+READOP                ; RX
29 CC032  000006 PREFETCH+READOP                ; HS
30 CC033  000006 PREFETCH+READOP                ; XS
31 CC034  000004 PREFETCH                       ; GG
32 CC035  000004 PREFETCH                       ; DI
33 CC036  000004 PREFETCH                       ; AP
34 CC037  000006 PREFETCH+READOP                ; UL
```

```
10003 .MAIN
01
02 CC040 000004 PREFETCH                    ; CI
03 CC041 000004 PREFETCH                    ; AC
04 CC042 000006 PREFETCH+READOP             ; NS
05 CC043 000006 PREFETCH+READOP             ; ND
06 CC044 000004 PREFETCH                    ; AS
07 CC045 000004 PREFETCH                    ; AD
08 CC046 000004 PREFETCH                    ; LS
09 CC047 000004 PREFETCH                    ; LD
10 CC050 000004 PREFETCH                    ; SH
11 CC051 000004 PREFETCH                    ; SL
12 CC052 000004 PREFETCH                    ; SE
13 CC053 000004 PREFETCH                    ; SN
14 CC054 000004 PREFETCH                    ; SO
15 CC055 000004 PREFETCH                    ; SZ
16 CC056 000004 PREFETCH                    ; SX
17 CC057 000004 PREFETCH                    ; GP
18 CC060 000006 PREFETCH+READOP             ; FA
19 CC061 000006 PREFETCH+READOP             ; FS
20 CC062 000006 PREFETCH+READOP             ; FM
21 CC063 000004 PREFETCH                    ; UNASSIGNED
22 CC064 000006 PREFETCH+READOP             ; FD
23 CC065 000004 PREFETCH                    ; CF
24 CC066 000006 PREFETCH+READOP             ; DL
25 CC067 000004 PREFETCH                    ; DS
26 CC070 000006 PREFETCH+READOP             ; AA
27 CC071 000006 PREFETCH+READOP             ; SS
28 CC072 000004 PREFETCH                    ; DP
29 CC073 000004 PREFETCH                    ; MH
30 CC074 000004 PREFETCH                    ; LK
31 CC075 000006 PREFETCH+READOP             ; IX
32       000002 .DO 2                       ;
33 CC076 000004 PREFETCH                    ; UNASSIGNED
34 CC077 000004 PREFETCH                    ; UNASSIGNED
```

```
10004 .MAIN
01
02                .ENDC
03 CC100 000004 PREFETCH                        ; UNASSIGNED
04 CC101 000004 PREFETCH                        ; DO
05 CC102 000006 PREFETCH+READOP                 ; EL
06 CC103 000006 PREFETCH+READOP                 ; HL
07 CC104 000006 PREFETCH+READOP                 ; LA
08 CC105 000006 PREFETCH+READOP                 ; LO
09 CC106 000006 PREFETCH+READOP                 ; LX
10 CC107 000006 PREFETCH+READOP                 ; WA
11 CC110 000006 PREFETCH+READOP                 ; WS
12 CC111 000004 PREFETCH                        ; AM
13 CC112 000006 PREFETCH+READOP                 ; WM
14 CC113 000004 PREFETCH                        ; AL
15 CC114 000004 PREFETCH                        ; XI
16 CC115 000007 PREFETCH+READOP+JUMP            ; JL
17 CC116 000007 PREFETCH+READOP+JUMP            ; JD
18 CC117 000007 PREFETCH+READOP+JUMP            ; JE
19 CC120 000006 PREFETCH+READOP                 ; XL
20 CC121 000006 PREFETCH+READOP                 ; ES
21 CC122 000006 PREFETCH+READOP                 ; EA
22 CC123 000006 PREFETCH+READOP                 ; ZL
23 CC124 000006 PREFETCH+READOP                 ; RL
24 CC125 000004 PREFETCH                        ; SP
25 CC126 000006 PREFETCH+READOP                 ; RE
26 CC127 000004 PREFETCH                        ; RS
27 CC130 000006 PREFETCH+READOP                 ; WD
28 CC131 000006 PREFETCH+READOP                 ; RX
29 CC132 000006 PREFETCH+READOP                 ; HS
30 CC133 000006 PREFETCH+READOP                 ; XS
31 CC134 000004 PREFETCH                        ; GG
32 CC135 000004 PREFETCH                        ; DI
33 CC136 000004 PREFETCH                        ; AP
34 CC137 000006 PREFETCH+READOP                 ; UL
```

```
10005 .MAIN
01
02 00140 000004 PREFETCH                    ; CI
03 00141 000004 PREFETCH                    ; AC
04 00142 000006 PREFETCH+READOP             ; NS
05 00143 000006 PREFETCH+READOP             ; ND
06 00144 000004 PREFETCH                    ; AS
07 00145 000004 PREFETCH                    ; AD
08 00146 000004 PREFETCH                    ; LS
09 00147 000004 PREFETCH                    ; LD
10 00150 000004 PREFETCH                    ; SH
11 00151 000004 PREFETCH                    ; SL
12 00152 000004 PREFETCH                    ; SE
13 00153 000004 PREFETCH                    ; SN
14 00154 000004 PREFETCH                    ; SO
15 00155 000004 PREFETCH                    ; SZ
16 00156 000004 PREFETCH                    ; SX
17 00157 000004 PREFETCH                    ; GP
18 00160 000006 PREFETCH+READOP             ; FA
19 00161 000006 PREFETCH+READOP             ; FS
20 00162 000006 PREFETCH+READOP             ; FM
21 00163 000004 PREFETCH                    ; UNASSIGNED
22 00164 000006 PREFETCH+READOP             ; FD
23 00165 000004 PREFETCH                    ; CF
24 00166 000006 PREFETCH+READOP             ; DL
25 00167 000004 PREFETCH                    ; DS
26 00170 000006 PREFETCH+READOP             ; AA
27 00171 000006 PREFETCH+READOP             ; SS
28 00172 000004 PREFETCH                    ; DP
29 00173 000004 PREFETCH                    ; MH
30 00174 000004 PREFETCH                    ; LK
31 00175 000006 PREFETCH+READOP             ; IX
32       000002 .DO 2                       ;
33 00176 000004 PREFETCH                    ; UNASSIGNED
34 00177 000004 PREFETCH                    ; UNASSIGNED
```

```
10006 .MAIN
01
02                .ENDC
03        000002 .DO 2                          ; CONDITIONAL PREFETCH
04 00200  000000 0                              ; UNASSIGNED
05 00201  000004 PREFETCH                       ; DO
06 00202  000006 PREFETCH+READOP                ; EL
07 00203  000006 PREFETCH+READOP                ; HL
08 00204  000006 PREFETCH+READOP                ; LA
09 00205  000006 PREFETCH+READOP                ; LO
10 00206  000006 PREFETCH+READOP                ; LX
11 00207  000006 PREFETCH+READOP                ; WA
12 00210  000006 PREFETCH+READOP                ; WS
13 00211  000004 PREFETCH                       ; AM
14 00212  000006 PREFETCH+READOP                ; WM
15 00213  000004 PREFETCH                       ; AL
16 00214  000000 0                              ; XI
17 00215  000003 READOP+JUMP                    ; JL
18 00216  000003 READOP+JUMP                    ; JD
19 00217  000003 READOP+JUMP                    ; JE
20 00220  000006 PREFETCH+READOP                ; XL
21 00221  000006 PREFETCH+READOP                ; ES
22 00222  000006 PREFETCH+READOP                ; EA
23 00223  000006 PREFETCH+READOP                ; ZL
24 00224  000006 PREFETCH+READOP                ; RL
25 00225  000004 PREFETCH                       ; SP
26 00226  000002 READOP                         ; RE
27 00227  000004 PREFETCH                       ; RS
28 00230  000006 PREFETCH+READOP                ; WD
29 00231  000006 PREFETCH+READOP                ; RX
30 00232  000006 PREFETCH+READOP                ; HS
31 00233  000006 PREFETCH+READOP                ; XS
32 00234  000004 PREFETCH                       ; GG
33 00235  000004 PREFETCH                       ; DI
34 00236  000004 PREFETCH                       ; AP
35 00237  000006 PREFETCH+READOP                ; UL
```

```
10007 .MAIN
01
02 CC240 000004 PREFETCH                    ; CI
03 CC241 000004 PREFETCH                    ; AC
04 CC242 000006 PREFETCH+READOP             ; NS
05 CC243 000006 PREFETCH+READOP             ; ND
06 CC244 000004 PREFETCH                    ; AS
07 CC245 000004 PREFETCH                    ; AD
08 CC246 000004 PREFETCH                    ; LS
09 CC247 000004 PREFETCH                    ; LD
10 CC250 000004 PREFETCH                    ; SH
11 CC251 000004 PREFETCH                    ; SL
12 CC252 000004 PREFETCH                    ; SE
13 CC253 000004 PREFETCH                    ; SN
14 CC254 000004 PREFETCH                    ; SO
15 CC255 000004 PREFETCH                    ; SZ
16 CC256 000004 PREFETCH                    ; SX
17 CC257 000004 PREFETCH                    ; GP
18 CC260 000006 PREFETCH+READOP             ; FA
19 CC261 000006 PREFETCH+READOP             ; FS
20 CC262 000006 PREFETCH+READOP             ; FM
21 CC263 000000 0                           ; UNASSIGNED
22 CC264 000006 PREFETCH+READOP             ; FD
23 CC265 000004 PREFETCH                    ; CF
24 CC266 000006 PREFETCH+READOP             ; DL
25 CC267 000004 PREFETCH                    ; DS
26 CC270 000006 PREFETCH+READOP             ; AA
27 CC271 000006 PREFETCH+READOP             ; SS
28 CC272 000004 PREFETCH                    ; DP
29 CC273 000004 PREFETCH                    ; MH
30 CC274 000000 0                           ; LK
31 CC275 000002 READOP                      ; IX
32       000002 .DO 2                       ;
33 CC276 000000 0                           ; UNASSIGNED
34 CC277 000000 0                           ; UNASSIGNED
```

```
10008 .MAIN
01
02                .ENDC
03  00300 000000 0                          ; UNASSIGNED
04  00301 000004 PREFETCH                   ; DO
05  00302 000006 PREFETCH+READOP            ; EL
06  00303 000006 PREFETCH+READOP            ; HL
07  00304 000006 PREFETCH+READOP            ; LA
08  00305 000006 PREFETCH+READOP            ; LO
09  00306 000006 PREFETCH+READOP            ; LX
10  00307 000006 PREFETCH+READOP            ; WA
11  00310 000006 PREFETCH+READOP            ; WS
12  00311 000004 PREFETCH                   ; AM
13  00312 000006 PREFETCH+READOP            ; WM
14  00313 000004 PREFETCH                   ; AL
15  00314 000000 0                          ; XI
16  00315 000003 READOP+JUMP                ; JL
17  00316 000003 READOP+JUMP                ; JD
18  00317 000003 READOP+JUMP                ; JE
19  00320 000006 PREFETCH+READOP            ; XL
20  00321 000006 PREFETCH+READOP            ; ES
21  00322 000006 PREFETCH+READOP            ; EA
22  00323 000006 PREFETCH+READOP            ; ZL
23  00324 000006 PREFETCH+READOP            ; RL
24  00325 000004 PREFETCH                   ; SP
25  00326 000002 READOP                     ; RE
26  00327 000004 PREFETCH                   ; RS
27  00330 000006 PREFETCH+READOP            ; WD
28  00331 000006 PREFETCH+READOP            ; RX
29  00332 000006 PREFETCH+READOP            ; HS
30  00333 000006 PREFETCH+READOP            ; XS
31  00334 000004 PREFETCH                   ; GG
32  00335 000004 PREFETCH                   ; DI
33  00336 000004 PREFETCH                   ; AP
34  00337 000006 PREFETCH+READOP            ; UL
```

```
!0009 .MAIN
01
02 C0340 000004 PREFETCH                    ; CI
03 C0341 000004 PREFETCH                    ; AC
04 C0342 000006 PREFETCH+READOP             ; NS
05 C0343 000006 PREFETCH+READOP             ; ND
06 C0344 000004 PREFETCH                    ; AS
07 C0345 000004 PREFETCH                    ; AD
08 C0346 000004 PREFETCH                    ; LS
09 C0347 000004 PREFETCH                    ; LD
10 C0350 000004 PREFETCH                    ; SH
11 C0351 000004 PREFETCH                    ; SL
12 C0352 000004 PREFETCH                    ; SE
13 C0353 000004 PREFETCH                    ; SN
14 C0354 000004 PREFETCH                    ; SO
15 C0355 000004 PREFETCH                    ; SZ
16 C0356 000004 PREFETCH                    ; SX
17 C0357 000004 PREFETCH                    ; GP
18 C0360 000006 PREFETCH+READOP             ; FA
19 C0361 000006 PREFETCH+READOP             ; FS
20 C0362 000006 PREFETCH+READOP             ; FM
21 C0363 000000 0                           ; UNASSIGNED
22 C0364 000006 PREFETCH+READOP             ; FD
23 C0365 000004 PREFETCH                    ; CF
24 C0366 000006 PREFETCH+READOP             ; DL
25 C0367 000004 PREFETCH                    ; DS
26 C0370 000006 PREFETCH+READOP             ; AA
27 C0371 000006 PREFETCH+READOP             ; SS
28 C0372 000004 PREFETCH                    ; DP
29 C0373 000004 PREFETCH                    ; MH
30 C0374 000000 0                           ; LK
31 C0375 000002 READOP                      ; IX
32       000002 .DO 2                       ;
33 C0376 000000 0                           ; UNASSIGNED
34 C0377 000000 0                           ; UNASSIGNED
```

```
10010 .MAIN
01
02                .ENDC
03        000004  .DO 4                   ; PREFETCH ERROR
04  CC400 000000  0                       ; UNASSIGNED
05  CC401 000000  0                       ; DO
06  CC402 000002  READOP                  ; EL
07  CC403 000002  READOP                  ; HL
08  CC404 000002  READOP                  ; LA
09  CC405 000002  READOP                  ; LO
10  CC406 000002  READOP                  ; LX
11  CC407 000002  READOP                  ; WA
12  CC410 000002  READOP                  ; WS
13  CC411 000000  0                       ; AM
14  CC412 000002  READOP                  ; WM
15  CC413 000000  0                       ; AL
16  CC414 000000  0                       ; XI
17  CC415 000003  READOP+JUMP             ; JL
18  CC416 000003  READOP+JUMP             ; JD
19  CC417 000003  READOP+JUMP             ; JE
20  CC420 000002  READOP                  ; XL
21  CC421 000002  READOP                  ; ES
22  CC422 000002  READOP                  ; EA
23  CC423 000002  READOP                  ; ZL
24  CC424 000002  READOP                  ; RL
25  CC425 000000  0                       ; SP
26  CC426 000002  READOP                  ; RE
27  CC427 000000  0                       ; RS
28  CC430 000002  READOP                  ; WD
29  CC431 000002  READOP                  ; RX
30  CC432 000002  READOP                  ; HS
31  CC433 000002  READOP                  ; XS
32  CC434 000000  0                       ; GG
33  CC435 000000  0                       ; DI
34  CC436 000000  0                       ; AP
35  CC437 000002  READOP                  ; UL
```

```
10011 .MAIN
01
02 CC440 000000 0                       ; CI
03 CC441 000000 0                       ; AC
04 CC442 000002 READOP                  ; NS
05 CC443 000002 READOP                  ; ND
06 CC444 000000 0                       ; AS
07 CC445 000000 0                       ; AD
08 CC446 000000 0                       ; LS
09 CC447 000000 0                       ; LD
10 CC450 000000 0                       ; SH
11 CC451 000000 0                       ; SL
12 CC452 000000 0                       ; SE
13 CC453 000000 0                       ; SN
14 CC454 000000 0                       ; SO
15 CC455 000000 0                       ; SZ
16 CC456 000000 0                       ; SX
17 CC457 000000 0                       ; GP
18 CC460 000002 READOP                  ; FA
19 CC461 000002 READOP                  ; FS
20 CC462 000002 READOP                  ; FM
21 CC463 000000 0                       ; UNASSIGNED
22 CC464 000002 READOP                  ; FD
23 CC465 000000 0                       ; CF
24 CC466 000002 READOP                  ; DL
25 CC467 000000 0                       ; DS
26 CC470 000002 READOP                  ; AA
27 CC471 000002 READOP                  ; SS
28 CC472 000000 0                       ; DP
29 CC473 000000 0                       ; MH
30 CC474 000000 0                       ; LK
31 CC475 000002 READOP                  ; IX
32       000002 .DO 2                   ;
33 CC476 000000 0                       ; UNASSIGNED
34 CC477 000000 0                       ; UNASSIGNED
```

```
10012 .MAIN
01                .ENDC
02 CC500 000000 0                    ; UNASSIGNED
03 CC501 000000 0                    ; DO
04 CC502 000002 READOP               ; EL
05 CC503 000002 READOP               ; HL
06 CC504 000002 READOP               ; LA
07 CC505 000002 READOP               ; LO
08 CC506 000002 READOP               ; LX
09 CC507 000002 READOP               ; WA
10 CC510 000002 READOP               ; WS
11 CC511 000000 0                    ; AM
12 CC512 000002 READOP               ; WM
13 CC513 000000 0                    ; AL
14 CC514 000000 0                    ; XI
15 CC515 000003 READOP+JUMP          ; JL
16 CC516 000003 READOP+JUMP          ; JD
17 CC517 000003 READOP+JUMP          ; JE
18 CC520 000002 READOP               ; XL
19 CC521 000002 READOP               ; ES
20 CC522 000002 READOP               ; EA
21 CC523 000002 READOP               ; ZL
22 CC524 000002 READOP               ; RL
23 CC525 000000 0                    ; SP
24 CC526 000002 READOP               ; RE
25 CC527 000000 0                    ; RS
26 CC530 000002 READOP               ; WD
27 CC531 000002 READOP               ; RX
28 CC532 000002 READOP               ; HS
29 CC533 000002 READOP               ; XS
30 CC534 000000 0                    ; GG
31 CC535 000000 0                    ; DI
32 CC536 000000 0                    ; AP
33 CC537 000002 READOP               ; UL
```

```
!0013 .MAIN
01
02 CC540 000000 0                    ; CI
03 CC541 000000 0                    ; AC
04 CC542 000002 READOP               ; NS
05 CC543 000002 READOP               ; ND
06 CC544 000000 0                    ; AS
07 CC545 000000 0                    ; AD
08 CC546 000000 0                    ; LS
09 CC547 000000 0                    ; LD
10 CC550 000000 0                    ; SH
11 CC551 000000 0                    ; SL
12 CC552 000000 0                    ; SE
13 CC553 000000 0                    ; SN
14 CC554 000000 0                    ; SO
15 CC555 000000 0                    ; SZ
16 CC556 000000 0                    ; SX
17 CC557 000000 0                    ; GP
18 CC560 000002 READOP               ; FA
19 CC561 000002 READOP               ; FS
20 CC562 000002 READOP               ; FM
21 CC563 000000 0                    ; UNASSIGNED
22 CC564 000002 READOP               ; FD
23 CC565 000000 0                    ; CF
24 CC566 000002 READOP               ; DL
25 CC567 000000 0                    ; DS
26 CC570 000002 READOP               ; AA
27 CC571 000002 READOP               ; SS
28 CC572 000000 0                    ; DP
29 CC573 000000 0                    ; MH
30 CC574 000000 0                    ; LK
31 CC575 000002 READOP               ; IX
32       000002 .DO 2                ;
33 CC576 000000 0                    ; UNASSIGNED
34 CC577 000000 0                    ; UNASSIGNED
```

```
IC014 .MAIN
01                .ENDC
02 CC600 000000 0                    ; UNASSIGNED
03 CC601 000000 0                    ; DO
04 CC602 000002 READOP               ; EL
05 CC603 000002 READOP               ; HL
06 CC604 000002 READOP               ; LA
07 CC605 000002 READOP               ; LO
08 CC606 000002 READOP               ; LX
09 CC607 000002 READOP               ; WA
10 CC610 000002 READOP               ; WS
11 CC611 000000 0                    ; AM
12 CC612 000002 READOP               ; WM
13 CC613 000000 0                    ; AL
14 CC614 000000 0                    ; XI
15 CC615 000003 READOP+JUMP          ; JL
16 CC616 000003 READOP+JUMP          ; JD
17 CC617 000003 READOP+JUMP          ; JE
18 CC620 000002 READOP               ; XL
19 CC621 000002 READOP               ; ES
20 CC622 000002 READOP               ; EA
21 CC623 000002 READOP               ; ZL
22 CC624 000002 READOP               ; RL
23 CC625 000000 0                    ; SP
24 CC626 000002 READOP               ; RE
25 CC627 000000 0                    ; RS
26 CC630 000002 READOP               ; WD
27 CC631 000002 READOP               ; RX
28 CC632 000002 READOP               ; HS
29 CC633 000002 READOP               ; XS
30 CC634 000000 0                    ; GG
31 CC635 000000 0                    ; DI
32 CC636 000000 0                    ; AP
33 CC637 000002 READOP               ; UL
```

```
10015 .MAIN
01
02 CC640 000000 0                    ; CI
03 CC641 000000 0                    ; AC
04 CC642 000002 READOP               ; NS
05 CC643 000002 READOP               ; ND
06 CC644 000000 0                    ; AS
07 CC645 000000 0                    ; AD
08 CC646 000000 0                    ; LS
09 CC647 000000 0                    ; LD
10 CC650 000000 0                    ; SH
11 CC651 000000 0                    ; SL
12 CC652 000000 0                    ; SE
13 CC653 000000 0                    ; SN
14 CC654 000000 0                    ; SO
15 CC655 000000 0                    ; SZ
16 CC656 000000 0                    ; SX
17 CC657 000000 0                    ; GP
18 CC660 000002 READOP               ; FA
19 CC661 000002 READOP               ; FS
20 CC662 000002 READOP               ; FM
21 CC663 000000 0                    ; UNASSIGNED
22 CC664 000002 READOP               ; FD
23 CC665 000000 0                    ; CF
24 CC666 000002 READOP               ; DL
25 CC667 000000 0                    ; DS
26 CC670 000002 READOP               ; AA
27 CC671 000002 READOP               ; SS
28 CC672 000000 0                    ; DP
29 CC673 000000 0                    ; MH
30 CC674 000000 0                    ; LK
31 CC675 000002 READOP               ; IX
32       000002 .DO 2                ;
33 CC676 000000 0                    ; UNASSIGNED
34 CC677 000000 0                    ; UNASSIGNED
```

```
10016 .MAIN
01                .ENDC
02 CC700 000000 0                    ; UNASSIGNED
03 CC701 000000 0                    ; DO
04 CC702 000002 READOP               ; EL
05 CC703 000002 READOP               ; HL
06 CC704 000002 READOP               ; LA
07 CC705 000002 READOP               ; LO
08 CC706 000002 READOP               ; LX
09 CC707 000002 READOP               ; WA
10 CC710 000002 READOP               ; WS
11 CC711 000000 0                    ; AM
12 CC712 000002 READOP               ; WM
13 CC713 000000 0                    ; AL
14 CC714 000000 0                    ; XI
15 CC715 000003 READOP+JUMP          ; JL
16 CC716 000003 READOP+JUMP          ; JD
17 CC717 000003 READOP+JUMP          ; JE
18 CC720 000002 READOP               ; XL
19 CC721 000002 READOP               ; ES
20 CC722 000002 READOP               ; EA
21 CC723 000002 READOP               ; ZL
22 CC724 000002 READOP               ; RL
23 CC725 000000 0                    ; SP
24 CC726 000002 READOP               ; RE
25 CC727 000000 0                    ; RS
26 CC730 000002 READOP               ; WD
27 CC731 000002 READOP               ; RX
28 CC732 000002 READOP               ; HS
29 CC733 000002 READOP               ; XS
30 CC734 000000 0                    ; GG
31 CC735 000000 0                    ; DI
32 CC736 000000 0                    ; AP
33 CC737 000002 READOP               ; UL
```

```
10017 .MAIN
01
02 CC740 000000 0                       ; CI
03 CC741 000000 0                       ; AC
04 CC742 000002 READOP                  ; NS
05 CC743 000002 READOP                  ; ND
06 CC744 000000 0                       ; AS
07 CC745 000000 0                       ; AD
08 CC746 000000 0                       ; LS
09 CC747 000000 0                       ; LD
10 CC750 000000 0                       ; SH
11 CC751 000000 0                       ; SL
12 CC752 000000 0                       ; SE
13 CC753 000000 0                       ; SN
14 CC754 000000 0                       ; SO
15 CC755 000000 0                       ; SZ
16 CC756 000000 0                       ; SX
17 CC757 000000 0                       ; GP
18 CC760 000002 READOP                  ; FA
19 CC761 000002 READOP                  ; FS
20 CC762 000002 READOP                  ; FM
21 CC763 000000 0                       ; UNASSIGNED
22 CC764 000002 READOP                  ; FD
23 CC765 000000 0                       ; CF
24 CC766 000002 READOP                  ; DL
25 CC767 000000 0                       ; DS
26 CC770 000002 READOP                  ; AA
27 CC771 000002 READOP                  ; SS
28 CC772 000000 0                       ; DP
29 CC773 000000 0                       ; MH
30 CC774 000000 0                       ; LK
31 CC775 000002 READOP                  ; IX
32       000002 .DO 2                   ;
33 CC776 000000 0                       ; UNASSIGNED
34 CC777 000000 0                       ; UNASSIGNED
```

```
10018 .MAIN
01                .ENDC
02                .END

0000 SOURCE LINES IN ERROR
```

RETURN LETTER

Title: CPU823, Microprogram. RCSL No.: 991 10172

A/S Regnecentralen af 1979/RC Computer A/S maintains a continual effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback, your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability:

Do you find errors in this manual? If so, specify by page.

How can this manual be improved?

Other comments?

Name: _____ Title: _____

Company: _____

Address: _____

Date: _____

Thank you

RCSL Nr. 46-F 0093

..... Fold here .....

..... Do not tear - Fold here and staple .....

Affix
postage
here

REGNECENTRALEN
af 1979

Information Department
Lautrupbjerg 1
DK-2750 Ballerup
Denmark