

Saturn Assembler SYSTEM LEX file <850429.0820> Mon Apr 29, 1985 8:21 am
Ver. 3.39/Rev. 2306 Page 1

```
1           TITLE SYSTEM LEX file <850429.0820>
2           *
3           * File Header
4 00000 3595      NIBASC \SYSTEMFN\     File Name
5445
54D4
64E4
5 00010 0000      CON(4) =fLEX        File Type
6 00014 00        NIBHEX 00          Flags
7 00016 0021      NIBHEX 0021        Time
8 0001A 2240      NIBHEX 224058       Date
58
9 00020 7050      REL(5) FILEND      File Length
0
10          *
11 00025 25        CON(2) #52         Id
12 00027 F5        CON(2) #5F         Lowest Token
13 00029 06        CON(2) #60         Highest Token
14 0002B 0000      NIBHEX 00000      Chain to next LEX table.
0
15          *
16 00030 F          NIBHEX F          Speed table omitted
17 00031 0200      CON(4) (TxTbSt)+1-(*) Offset to text table
18 00035 0000      CON(4) 0            Offset to message table
19 00039 9920      REL(5) POLLhn      Offset to poll handler
0
20          *
21          * Main Table
22 0003E 000      CON(3) 0            5F SYSTEM$ 
23 00041 5300      REL(5) XSyst$ 
0
24 00046 F          NIBHEX F          Function.
25
26 00047 110      CON(3) 17          60 SYSTEM
27 0004A 7300      REL(5) XSystm
0
28 0004F F          NIBHEX F          Function.
29
30          * Text Table
31 00050 D          TxTbSt NIBHEX D
32 00051 3595      NIBASC \SYSTEM$\ 
3545
54D4
42
33 0005F F5        CON(2) #5F
34
35 00061 B          NIBHEX B
36 00062 3595      NIBASC \SYSTEM\ 
3545
54D4
37 0006E 06        CON(2) #60
38
39 00070 1FF        NIBHEX 1FF       Text termination
```

Saturn Assembler SYSTEM LEX file <850429.0820> Mon Apr 29, 1985 8:21 am
Ver. 3.39/Rev. 2306 Keyword execution Page 2

```
40           STITLE Keyword execution
41           ****
42           ****
43           **
44           ** Name: XSystm , XSyst$ 
45           **
46           ** Category: LOCAL (Function Execute)
47           **
48           ** Purpose: Execute SYSTEM and SYSTEM$ functions.
49           **
50           ** Entry: From EXPR.
51           ** Exit: To FNRTN1 or STR$.
52           **
53           ** Calls: POPMTH,POP1R,FLTDH,FPOLL,TBLJMP,D0=AVS, CONVUC
54           ** HXDCW,FLOAT,STRHDR,MOVEU0,REV$,TBLJM2,FNRTN1,STR$,MFERR
55           **
56           ** Uses: A-D, D0, D1, P, S0-S7, R0, R1, R2, R3
57           **
58           ** Stk lvs: Expression execution can use 4.
59           **
60           ** NOTES:
61           ** The two functions, SYSTEM and SYSTEM$, are generic peek-
62           ** type functions for returning system settings. A general
63           ** description of the design of these two keywords follows.
64           ** Details are in the poll interface (pSysFn), below.
65           **
66           ** Syntax:
67           ** SYSTEM( <type$> )
68           ** Returns a numeric value corresponding to the setting
69           ** of the <type$> string.
70           **
71           ** SYSTEM$( <type$> | <arg#> )
72           ** Returns:
73           ** 1) A string value corresponding to the setting of
74           ** the <type$> string.
75           ** 2) A string which is a valid argument for the SYSTEM/
76           ** SYSTEM$ fcns, or the null string. SYSTEM$(<arg#>)
77           ** returns the nth tabled argument for the SYSTEM/
78           ** SYSTEM$ fcns, where 'n' is the rounded integer
79           ** value of <arg#>. An extra character is inserted
80           ** in the first position of the returned string:
81           **   1st char meaning
82           ** -----
83           **   n    valid arg for SYSTEM only.
84           **   s    valid arg for SYSTEM$ only.
85           **   <blank> valid arg for both
86           **   <null>  no corresponding argument
87
88           ** The numeric argument in SYSTEM$ allows the user to generate
89           ** a list of valid arguments for the SYSTEM/SYSTEM$ fcns.
90           **
91           ** SYSTEM$ is designed to accept all SYSTEM arguments, in
92           ** effect performing a STR$(SYSTEM(<type$>)). This is done
93           ** automatically, provided the LEX file designates the argument
94           ** as acceptable for the SYSTEM$ function. For this
```

```

95      ** reason, only in exceptional cases should an argument be
96      ** specified as "numeric only."
97      **
98      ** For example,
99      -- After the user set WIDTH 80, SYSTEM("WIDTH") would
100     ** return the value 80. SYSTEM$("WIDTH") returns "80".
101     ** Set LOCK 'Juju'; SYSTEM$("PASSWORD") returns "Juju".
102     ** Depending on the ordering in memory of LEX files which
103     ** answer the pSysFn poll, SYSTEM$(7) might return the
104     ** string "sENDLINE". The 's' indicates that "ENDLINE" is
105     ** a valid argument only for the string function SYSTEM$.
106     ** SYSTEM$(-5) for example, would return the null string,
107     ** as there is no corresponding tabled argument for that
108     ** number. SYSTEM$(1E6) would return the null string, too.
109     ** -- This program would display all system settings (pro-
110     ** vided no LEX file has any 'n' type arguments):
111     **   10 I=1
112     **   20 S$=SYSTEM$(I)[2] @ IF S$="" THEN STOP
113     **   30 DISP S$,SYSTEM$(S$) @ I=I+1 @ GOTO 20
114     **
115     ** If the main execution routine is already in memory, all
116     ** that is needed to add new arguments to the SYSTEM/SYSTEM$  

117     ** functions is a pSysFn poll handler. In fact, this master
118     ** file answers its own poll to implement the following list
119     ** of arguments ('s'=string; 'n/s'=numeric or string):
120     **   WIDTH (n/s) WIDTH setting.
121     **   PWIDHT (n/s) PWIDHT setting.
122     **   PASSWORD (s) Current LOCK password.
123     **   CMDSTK (n/s) #levels in command stack.
124     **   KEYCOUNT (n/s) #keys in key buffer.
125     **   CLOCK (n/s) System clock speed.
126     **   ENDLINE (s) ENDLINE setting.
127     **   DISPFORM (s) Display format.
128     **   CONTRAST (n/s) CONTRAST setting.
129     **   LDELAY (n/s) Line DELAY rate (1st DELAY param).
130     **   CDELAY (n/s) Char DELAY rate (2nd DELAY param).
131     **
132     ** If the main execution routine is not in memory, use this
133     ** LEX file and replace the poll handler with an appropriate
134     ** one to handle your particular argument. Or the handler
135     ** can simply be expanded to include new arguments.
136     **
137     ** The main execution routine does four things:
138     1) Pops the argument, strips out blanks and converts all
139     ** characters to upper case. It stores this 'stripped'
140     ** version in memory without touching the original arg.
141     2) Stores pointers in R registers, then generates a fast
142     ** poll (pSysFn) for LEX files to 'handle' the arg.
143     3) It provides a table scan routine which chains through
144     ** each LEX file's SYSTEM/SYSTEM$ argument table, check-
145     ** ing for a match. If a match is found, it jumps to
146     ** the LEX file processor (address is found in table).
147     4) Accepts the returned function value, converts it (if
148     ** necessary), places it on the math stack and exits.
149

```

```

150      ** The simplest implementation of the SYSTEM/SYSTEM$ func-
151      ** tions is through a table of valid arguments. Each poll
152      ** handler passes the address of its table to the table scan
153      ** routine, and the table is searched for a match with the
154      ** argument. Only COMPLETE matches are accepted; partial
155      ** matches are not. Tabled strings are limited in that:
156      **   -- they must be uppercase only.
157      **   -- they cannot include blanks.
158      **   -- they must have 1 through 16 characters.
159      **   -- they cannot contain characters after a colon (:').
160      **
161      ** Tables will satisfy almost all applications. For those
162      ** applications which accept lower case, blanks or more than
163      ** 16 characters, you need to 'parse' the string argument
164      ** individually to check for a match. Applications which
165      ** allow slight variations in a string argument can use the
166      ** ':' feature to allow the user to extend the argument.
167      ** Characters after a ':' are ignored by the main routines,
168      ** so the LEX file must 'parse' the extra characters itself.
169      **
170      ** Examples of tabled arguments for SYSTEM/SYSTEM$:
171      ** argument can it be tabled?
172      ** -----
173      ** DISPLAYIS Yes.
174      ** DISPLAY IS No -- blanks not allowed.
175      ** CHECK<FILES Yes.
176      ** 123GO! Yes.
177      ** Variables No -- cannot contain lower case.
178      ** STATISTICSVARIABLES No -- must be <=16 chars.
179      ** BUFSIZE: Yes (any characters after the ':'
180      ** would be 'parsed' by the LEX file handler).
181      ** VARIABLES:CTHROUGHQ No -- no chars after a colon.
182      **
183      ** In the examples which are invalid for table entries, your
184      ** application can still accept them if you do the work.
185      ** Note that the user-supplied argument is stripped of
186      ** blanks and converted to upper case, so the user is not
187      ** constrained by the above rules for table entries. For
188      ** example, the user might specify "d I S pla y i S"
189      ** for his argument, and it would match on "DISPLAYIS".
190      **
191      ** Whether the poll handler uses the table scan feature or
192      ** parses the argument itself, it has the responsibility of
193      ** computing the function value. The processor routine com-
194      ** putes the value, and returns from the poll with the value
195      ** in the A register (details in the pSysFn interface). For
196      ** the SYSTEM function, the returned value must be numeric.
197      ** For SYSTEM$, the value can be string or numeric; numeric
198      ** values are automatically converted using STR$.
199      **
200      ** Detail:
201      ** Storage of pointers (for string argument case):
202      **   R0(A)=PC (D0 at entry to function execute).
203      **   R1(A)=address past string argument.
204      **   R2(A)=address of stripped string argument.

```

Saturn Assembler SYSTEM LEX file <850429.0820> Mon Apr 29, 1985 8:21 am
Ver. 3.39/Rev. 2306 Keyword execution

```
205      ** R3(A)=address of TblScn routine (for pSysFn poll).
206      **
207      ** RAM usage (for string argument case):
208      ** At entry to SYSTEM/SYSTEM$ function:
209      ** +-----+
210      ** | header: string chars |
211      ** +-----+
212      ** ^   ^
213      ** D1 string argument AvMemEnd
214      **
215      ** The string argument is popped, blanks stripped off, the
216      ** characters converted to uppercase, and the 'stripped'
217      ** string written in memory 32 nibs below the argument's
218      ** header. The string argument itself is left untouched.
219      ** At the time of the pSysFn poll (label SysPol):
220      ** +-----+
221      ** | stripped strg | header: string chars |
222      ** +-----+
223      ** ^   ^
224      ** Addr in R2(A) string arg (unchanged) Addr in R1(A)
225      **
226      ** History:
227      ** Date    Programmer        Modification
228      ** -----
229      ** 04/17/85  MB            Coded
230      ****
231      *
232      ****
233      *
234      ****
235      *
236      ** Name: (S) pSysFn
237      **
238      ** Category: POLL
239      **
240      ** Type:     FPOLL
241      **
242      ** Purpose: To extend SYSTEM/SYSTEM$ function argument list.
243      **
244      ** Should poll be "Handled" (return with XM=0)?: Yes.
245      **
246      ** Meaning of "Handling" Poll (what does code do if handled?):
247      ** Argument is valid, was handled properly, and the function
248      ** value has been computed. Otherwise, return "not handled".
249      **
250      ** Entry conditions for handler (registers, ST, RAM, etc.):
251      ** Carry set (fast poll); B[A]=Poll#(44h); HEX mode; P=0.
252      ** See Xsystm description for RAM contents.
253      ** Other details provided in this full explanation:
254      **
255      ** This poll is issued by the SYSTEM/SYSTEM$ functions. Its
256      ** main purpose is to ask poll handlers to pass back the ad-
257      ** dress of a table which is used to scan for a match with
258      ** the function argument. Most applications will depend com-
259      ** pletely on the table scan for parsing; special cases (en-
```

Saturn Assembler SYSTEM LEX file <850429.0820> Mon Apr 29, 1985 8:21 am
Ver. 3.39/Rev. 2306 Keyword execution

```
260      ** tries>16 chars, containing blanks or lower case chars or
261      ** otherwise requiring special parsing) are covered below.
262      **
263      ** There should be three parts to a pSysFn poll handler.
264      ** PART I: Initial poll interception. Put the address of
265      ** your SYSTEM/SYSTEM$ table in RSTK, jump to the table
266      ** scanner, whose address is in R3(A).
267      ** PART II: Table and associated processor routines. The
268      ** table construction is explained below. Each entry
269      ** has an associated processor routine to compute the
270      ** function value, and exit the poll.
271      ** PART III: Return from table scanner. If no match was
272      ** found in your table, the scanner returns for further
273      ** processing; this code is usually a simple RTNSXM.
274      **
275      ** PART I: Initial poll interception.
276      ** ----- Register contents -----
277      ** | If a string argument was specified:
278      ** | S2=0.
279      ** | S0=1 iff SYSTEM$ function is being executed.
280      ** | S1=1 iff SYSTEM function is being executed.
281      ** | S3=1 iff the argument <=16 non-blank characters.
282      ** | S4 through S7 (one nibble)=length of arg= #chars-1.
283      ** | R2(A)= address of stripped string (32 nibbles below
284      ** | the header of the string argument itself).
285      **
286      ** If a numeric argument was specified, SYSTEM$(<arg#>):
287      ** | S2=1. S0=1, S1=0, S3=1. S4-S7 all=0.
288      ** | R2(A)=counter (in hex) used to identify the nth valid
289      ** | argument. Initial value=<arg#>-1. It is decremented
290      ** | by 1 for each string found; when it carries, the
291      ** | current string is returned as the function value.
292      **
293      ** R0(A)= PC (D0 at entry to function).
294      ** | R1(A)= address of top of math stack (past string arg).
295      ** | R3(A)= address of TblScn routine (table scanner).
296      ** -----
297      **
298      ** PART II: Match is found in your table, jump to processor.
299      ** ----- Register contents -----
300      ** | Same register conditions as in Part I. In addition,
301      ** | D1 points to top of math stack (from R1); A(W)=0; D0
302      ** | has been loaded with nibbles found in the table entry.
303      **
304      ** If your matched entry includes a colon (':'), you need
305      ** to parse the extra chars, and act on them accordingly.
306      **
307      ** The processor is not invoked if S2=1. If the proper
308      ** entry was found in your table for SYSTEM$(<arg#>), the
309      ** main execution routines use the tabled string as the
310      ** return function value, and exit immediately.
311      **
312      ** PART III: No match in your table, return for clean-up.
313      ** ----- Register contents -----
314      ** | Same register conditions as in Part I.
```

```

315      ** +-----+
316      ** Part III is NOT performed if S3=0 (if the string arg
317      ** >16 chars.) Poll continues immediately, "Not Handled."
318      **
319      ** [SPECIAL CASES (valid arguments which cannot be tabled):
320      ** [ All entry conditions and comments, above, for the three
321      ** [ parts apply. If your LEX file allows any special case
322      ** [ arguments, you should 'parse' the argument immediately
323      ** [ in Part I. After checking the argument, if you have a
324      ** [ table you can jump to the table scanner as usual. Note
325      ** [ that if S3=0 (argument >16 chars), no table scan will
326      ** [ be performed; you have no further chance at execution.
327      ** [ The first thing to do in Part I is to check S2: if S2=1
328      ** [ (SYSTEM$(<arg#>)), get the counter in R2(A). If you
329      ** [ have, say, six non-tabled valid arguments, decrement
330      ** [ this counter by 6 (in hex mode) and replace it in R2.
331      ** [ If it carries, then return the appropriate string as
332      ** [ the final value. If S2=0, parse the argument yourself
333      ** [ and check for a match with your list. If a match, com-
334      ** [ pute the final function value and return.
335      ** [ If you have no table, Parts II and III will be omitted.
336      ** [ If you have a table, Part III will return (only if no
337      ** [ match), and you have a chance to clean up anything.
338      **
339      ** Some examples to make the process clear:
340      1) SYSTEM("WIDTH") issues a pSysFn poll. A handler passes
341      back the address of a table; WIDTH matches to an entry
342      in the table; the table scanner jumps to the processor,
343      which computes the function value and returns.
344      2) SYSTEM("Buf size:133") converts argument to "BUFSIZE:"
345      by stripping blanks and converting to upper case. It
346      issues a poll, scans tables, matches on "BUFSIZE:",
347      jumps to the processor. The processor 'parses' the
348      value 133, computes the function value and returns.
349      3) SYSTEM$(6) sets S2=1 to indicate a numeric arg, issues
350      a pSysFn poll. It should return the 6th valid string
351      argument for the SYSTEM/SYSTEM$ fcns (the 6th argument
352      is arbitrary, as it depends on the ordering of the LEX
353      files, as well as the ordering of the tables). Each LEX
354      file passes back the address of its table; the table
355      scanner chains through, decrementing the counter for
356      each entry. When the counter carries, the current table
357      entry is returned as the function value.
358      SPECIAL CASES:
359      4) SYSTEM$("STATISTICSVARIABLES") issues a pSysFn poll. A
360      handler determines immediately that the argument is
361      >16 chars (S3=0) and, since it has several legal argu-
362      ments of >16 chars, it 'parses' the argument itself.
363      It finds a match with this argument, processes the
364      function itself, and returns through the poll handler.
365      5) A certain pSysFn poll handler accepts "VarA" through
366      "VarZ", without using the ':' feature (just hypothetic-
367      al...). Instead of 26 table entries, it only uses one:
368      it intercepts the pSysFn poll, decreases the argument
369      length by 1, jumps to the table scanner. If a match is

```

```

370      ** not made on "VAR", it returns to Part III of the poll
371      ** handler; the poll handler corrects the argument length
372      ** by adding 1, and lets the poll continue "not handled".
373      **
374      ** Note that in example 3, any LEX file with special cases
375      ** must decrement the counter in R2(A) by the number of spec-
376      ** ial cases it handles; if it carries, the LEX file must re-
377      ** turn a string as the function value.
378      **
379      ** Details on table construction:
380      ** The first nibble of a table entry is for characterization:
381      ** 0= End of table, Part III code follows (usually RTNSXM).
382      ** 9= Entry is valid for SYSTEM$ only.
383      ** A= Entry is valid for SYSTEM only.
384      ** B= Entry is valid for both SYSTEM and SYSTEM$.
385      ** No other values allowed for the characterization nibble.
386      ** It would be extremely rare to have a value of 'A', since
387      ** the SYSTEM$ function is designed to perform a STR$ on
388      ** numeric values -- but only if the low bit is set!
389      **
390      ** The 2nd nibble of a table entry is the length: #chars -1.
391      ** Next follows the ASCII characters for the valid argument.
392      **
393      ** The next 5 nibbles are (usually) the address where the
394      ** system setting will be read; these nibbles are placed in
395      ** D0 before jumping to the processor. (These nibbles can
396      ** be anything. They are put into D0, to use as you like.)
397      **
398      ** Next follows a 3 nibble relative offset to a processor
399      ** routine, invoked if a match is made on the table entry.
400      **
401      ** A complete pSysFn poll handler might look like this
402      ** (simple table scan, no special cases):
403      **      GOSUB A3          Addr of table into RSTK.
404      **      CON(1) #B          Numeric or string fcn.
405      **      CON(1) ((A2)-(A1))/2-1 Length of string.
406      **      A1    NIBASC \CONTRAST\ ASCII chars.
407      **      A2    CON(5) #2E3FE   Addr of CONTRAST setting.
408      **      REL(3) A4          Offset to processor.
409      **      *
410      **      CON(1) 0          End of table.
411      **      RTNSXM           No match, "not handled".
412      **      *
413      **      A3    C=R3          Fetch addr of TblScn.
414      **      RSTK=C
415      **      RTNCC
416      **      *
417      **      A4    A=DATO 1      Read contrast nibble.
418      **      P=      1          "Hex value in A(W)".
419      **      XM=0            "Poll handled".
420      **      RTNCC           Return function value.
421      **      *
422      **      Normal exit conditions from handler if handled:
423      **      HEX mode, XM=0, S0-S7 unchanged, R0(A) unchanged.
424      **      Value of P determines where/what the returned value is:

```

```

425      ** P=0: If a numeric value and A(W) contains the float-
426      ** ing point decimal value; or
427      ** P=0: If a string function, the string characters have
428      ** been placed on the stack, and A(W) contains the
429      ** string header. R1(A) points to the first string
430      ** character on the stack (i.e., low address). Or
431      ** P=0: If a complex value, the value has been placed on
432      ** the stack, D1=16 nibs past '0E' byte, and A(W)
433      ** contains '0E' and low 14 nibs of imaginary part.
434      ** P=1: If a numeric value and A(W) contains the value
435      ** in HEX. It will be converted into decimal.
436      ** P=2: If a string function, B(W)=#nibs in the string,
437      ** A(A)=address of the string chars, and the chars
438      ** are to be moved to the stack and reversed.
439      ** P=3: Same as case P=2, except no string reversal.
440      **
441      ** Except for the case P=0 for string or complex, the ad-
442      ** dress in R1(A) should be the same as at entry. For
443      ** cases P=2 and P=3, the entire B register is used for a
444      ** nibble count; so all upper nibbles should be=0. Sim-
445      ** ilarly for case P=1, the upper nibbles of A should be=0.
446      ** You can always return a numeric value for the SYSTEM$  

447      ** function, as it will be converted automatically into a
448      ** string using STR$. (But the characterization nibble for
449      ** the table entry must be 9 or B; a value of A will not ac-
450      ** cept the argument for the string function.) A type 'B'
451      ** argument can return different numeric and string values,
452      ** if you like. E.g., SYSTEM("LOOP") might return the value
453      ** 2, but SYSTEM$("LOOP") might return the string "BUSY".
454      **
455      ** [SPECIAL CASES:  

456      ** [ S2 is used to distinguish the case SYSTEM$(<arg#>) from
457      ** [ a string argument. If the poll is handled with case P=2
458      ** [ and S2=1, the first character in the string will be re-
459      ** [ placed with an 'n', 's', or <blank>, in this manner:
460      ** [ first byte in string           first byte replaced
461      ** [ (bit values. x=don't care)   with character:  

462      ** -----
463      ** [     xxxx x001             s
464      ** [     xxxx x010             n
465      ** [     xxxx x011             <blank>
466      ** [ other values            random character
467      ** [ So if S2=1 and you return with P=2, your string will be
468      ** [ altered. Never return S2=1, P=2 and B=0 (null string),
469      ** [ as the first byte past the top of the math stack will be
470      ** [ overwritten with 'n', 's' or <blank>. (P=3 is OK.)
471      **
472      ** Normal exit conditions from handler if not handled:
473      **     HEX mode, XM=1. D(A) untouched!
474      **
475      ** Available subroutine levels: 2
476      **
477      ** NOTE: Except for an "Insufficient Memory" error, a poll
478      ** handler should always return through the poll routine.
479      ** You should not give an error (other than MemErr) when

```

```

480      ** processing the function. E.g., if "BUFSIZE:" is your
481      ** argument, and you expect a valid number after the colon,
482      ** don't error if the user types SYSTEM("BUFSIZE:MUCK").
483      ** You might evaluate "MUCK" (as in VAL("MUCK")), or return
484      ** a flag value (such as -31) to indicate an error. And/or
485      ** you might give a warning. But don't error out.
486      **
487      ** What registers/RAM may be used if handled?:
488      **     A-D, D0, D1, P, R2, R3, R4, R0(15-5), R1(15-5),
489      **     S8 through S11. Function scratch (FUNCRO, etc.).
490      **     Available mem below address in R1(A).
491      **     R1(A) may be adjusted to point to top of math stack.
492      **
493      ** What registers/RAM may be used if not handled?:
494      **     A-C, D(15-5), D0, D1, P, R0(15-5), R1(15-5), R2(15-5),
495      **     R3(15-5), R4, S8 through S11. Function scratch,
496      **     Avail Mem below: R2(A) if S2=0, R1(A) if S2=1.
497      **
498      ** Special memory/pointer considerations:
499      **     Poll handlers cannot move memory in the file chain area;
500      **     doing so may invalidate the address in R3(A).
501      **
502      ** History:
503      **     Date    Programmer        Modification
504      **     -----  -----
505      ** 04/20/85 MB    Implemented.
506      ****
507      ** 508      Systm$ EQU 0          "Executing SYSTEM$".
509      ** 509      System EQU 1          "Executing SYSTEM".
510      ** 510      NumArg EQU 2          "Numeric argument specified."
511      ** 511      Chrcnt EQU 3          "Character count".
512      ** 512      pSysFn EQU #44         Poll number.
513      ****
514      *-----*
515 00073 C11      NIBHEX C11      SYSTEM$(<type$|<arg#>)
516 00076 08       XSyst$ CLRST
517 00078 850      ST=1  Systm$  "Executing SYSTEM$".
518 00078 5A0      GONC  SYS01  (BET)
519      *-----*
520 0007E 411      NIBHEX 411     SYSTEM(<type$>)
521 00081 08       XSystm CLRST
522 00083 851      ST=1  System   "Executing SYSTEM".
523 00086 1574     SYS01  C=DAT1 S  Read data type from stack.
524 0008A 137      CD1EX
525 0008D 135      D1=C
526 00090 D7       D=C=A
527 00092 136      CDOEX
528 00095 108      R0=C
529 00098 8F00     GOSBVL =POPMTH (1B3DB) Pop item. D1 points past.
530 0009F DB       C=D  A       Restore D1, save new
531 000A1 137      CD1EX
532 000A4 109      R1=C
533 000A7 853      ST=1  Chrcnt "Counting chars".

```

Saturn Assembler SYSTEM LEX file <850429.0820> Mon Apr 29, 1985 8:21 am
Ver. 3.39/Rev. 2306 Keyword execution Page 11

```

534 000AA B46      C=C+1 S          String arg?
535 000AD 496      GOC Sxx03      Yes. C(S)=0.
536 000B0 852      ST=1 NumArg    " SYSTEM(<arg#>) ".
537 000B3 8F00     GOSBVL =POP1R (0E8FD) Pop real, error on complex.
      000
538 000BA 8F00     GOSBVL =FLTDH (1B223) Convert A(W) to hex.
      000
539 000C1 583      GONC NotFnd   Out of range. Null string.
540 000C4 CC       A=A-1 A          Base 0.
541 000C6 433      GOC NotFnd   Arg=0: null string.
542 000C9 102      R2=A          Save counter for poll.
543 000CC 68B0     GOTO Sxx23    Poll for table addresses.
544 *
545 000D0 07       Syspol C=RSTK  Address of TblScn.
546 000D2 10B      R3=C          Into R3 for polling.
547 000D5 8F00     GOSBVL =FPOLL (1250A)
      000
548 000DC 44       CON(2) =pSysFn
549 000DE 5B1      GONC NotFnd   NC=not handled.
550 000E1 119      SysP11 C=R1   Fetch math stack pointer,
551 000E4 135      D1=C          put in D1.
552 000E7 8F00     GOSBVL =TBLJMP (0242A) Jump on P.
      000
553 000EE 661      REL(3) HNDL=0 Dec value/string hdr in A.
554 000F1 251      REL(3) HNDL=1 Convert A into dec, put on stack.
555 000F4 981      REL(3) HNDL=2 Read chars from A(A) addr, REV$.
556 000F7 981      REL(3) HNDL=3 Same as '2', but don't reverse.
557 *
558 000FA AF0      NotFnd A=0 W
559 000FD A0C      A=1-P          A(W)=header for null string.
560 00100 872      ?ST=1 NumArg  SYSTEM$(<arg#>)?
561 00103 ED       GOYES SysP11 Yes. Return null string.
562 00105 3100    errarg LC(2) =eIVARG (#0B) No. "ERR:Invalid Arg".
563 00109 8D00    error GOVLNG =MFERR (09393)
      000
564 *
565 00110 3100    Memerr LC(2) =eMEM (#18)
566 00114 44F      GOC error    (BET)
567 *
568 00117 D8       Sxx03 B=A A          Length of string in nibs.
569 00119 1CF      D1=D1- 16
570 0011C 1CF      D1=D1- 16      32 nibs below strg hdr.
571 0011F 137      CD1EX        D1=past strg, C=32 below hdr.
572 00122 10A      R2=C          Save addr of stripped string.
573 00125 8F00     GOSBVL =D0=AVS (09B2C) Set A=D0=AvMemSt.
      000
574 0012C 8BE      ?A>=C A          Past AvMem?
575 0012F 1E       GOYES Memerr Yes. "Insufficient Memory".
576 00131 134      D0=C          D0=32 nibs below header.
577 00134 A4E      C=C-1 S          C(S)=F for counter.
578 00137 3102    Sxx05 LCASC \\ \ Count nibs in string.
579 0013B CD       Sxx07 B=B-1 A          String exhausted.
580 0013D 433      GOC Sxx19   (2*nibs=#bytes.)
581 00140 CD       B=B-1 A          To next char.
582 00142 1C1      D1=D1- 2

```

Saturn Assembler SYSTEM LEX file <850429.0820> Mon Apr 29, 1985 8:21 am
Ver. 3.39/Rev. 2306 Keyword execution Page 12

```

583 00145 14B      A=DAT1 B          Read next char.
584 00148 962      ?A=C B          Blank?
585 0014B 0F       GOYES Sxx07    Yes, strip off.
586 0014D B46      C=C+1 S          Count non-blank chars.
587 00150 5A0      GONC Sxx09    NC-less than 16.
588 00153 863      ?ST=0 Chrcnt  Initial character?
589 00156 32       GOYES Sxx21    No. >16 characters.
590 00158 843      ST=0 Chrcnt  Yes. "Counting chars".
591 0015B 8F00     Sxx09 GOSBVL =CONVUC (152AA) Convert to upper case.
      000
592 00162 148      DAT0=A B          Write out upper case char.
593 00165 161      DO=D0+ 2      Increment buffer pointer.
594 00168 31A3     LCASC :\ \
595 0016C 966      ?A#C B          ':' character?
596 0016F 8C       GOYES Sxx05    No. Keep scanning.
597 *               *               Yes. Ignore rest of arg.
598 00171 873      Sxx19 ?ST=1 Chrcnt Any non-blank chars?
599 00174 19       GOYES errarg  No. "Invalid Arg".
600 00176 853      ST=1 Chrcnt  "Argument <= 16 chars".
601 00179 09       Sxx21 C=ST    C=ST
602 0017B 80DF      P=C 15      Char count to P.
603 0017F 80F1      CPEX 1      Char count to C(1), P=0.
604 00183 0A       ST=C          Char count to ST(1).
605 00185 774F      Sxx23 GOSUB Syspol Get TblScn addr into R3.
606 *               *               !!! TblScn must follow immediately.
607 *
608 *               *               TblScn may be used by all pSysFn poll handlers. Since it
609 *               *               is a fast poll, D(A) cannot be used in this routine. The
610 *               *               address of TblScn is passed to the poll handlers in R3(A).
611 *
612 *               *               !!! "GOSUB Syspol" must precede TblScn.
613 00189 07       TblScn C=RSTK  Get address of table.
614 0018B 134      D1=C          D0=address of table.
615 0018E 873      ?ST=1 Chrcnt <= 16 chars?
616 00191 40       GOYES TblS03  Yes. Search table.
617 00193 00       RTNSXM    No, >16. Exit "Not Handled".
618 *
619 00195 11A      TblS03 C=R2    Address of stripped string.
620 00198 135      D1=C          D1=addr of stripped string.
621 0019B 09       C=ST          C(1)=#chars-1, S0&S1 needed.
622 0019D D5       B=C A          Save in B(B).
623 0019F 20       TblS05 P= 0      Characteriz'n for next entry.
624 001A1 14E      C=DATA0 B    End of table?
625 001A4 90A      ?C=0 P       Yes. Return to LEX file.
626 001A7 07       GOYES Return  SYSTEM$(<arg#>)?
627 001A9 862      ?ST=0 NumArg No.
628 001AC 80       GOYES TblS07  Yes. Count entries.
629 001AE 1C0      D1=D1- 1      Found it!
630 001B1 497      GOC Sfound  To tabled strg. Carry clear.
631 001B4 161      TblS07 D0=D0+ 2 If match hopeless.
632 001B7 D0       TblS09 A=0 A Cry set only from jump below.
633 001B9 433      GOC TblS17  Check function type.
634 001BC 0E05    C=C&B P       Right length, function type?
635 001C0 965      ?C#B B       No. A=0, match hopeless.
636 001C3 60       GOYES TblS11

```

Saturn Assembler SYSTEM LEX file <850429.0820> Mon Apr 29, 1985 8:21 am
Ver. 3.39/Rev. 2306 Keyword execution Page 13

```

637 001C5 1537 A=DAT1 W Yes. Read stripped string.
638 001C9 A66 TblS11 C=C+C B Recover #nibs in string.
639 001CC 8D01 P=C 1 P=#nibs-1 in string.
640 001D0 581 GONC TblS15 8 or less chars.
641 001D3 1567 C=DAT0 W More than 8. Read first 8.
642 001D7 16F D0=D0+ 16 To remaining chars.
643 001DA 976 ?A#C W Match in first 8?
644 001DD AD GOYES TblS09 No. Set A=0, match hopeless.
645 001DF 17F D1=D1+ 16 Yes. To remaining chars.
646 001E2 1531 A=DAT1 WP Read stripped chars.
647 001E6 1CF D1=D1- 16 Restore D1.
648 001E9 1561 TblS15 C=DAT0 WP Read remaining table chars.
649 001ED 136 TblS17 CDOEX Step D0 past tabled
650 001F0 809 C+P+1 characters.
651 001F3 136 CDOEX
652 001F6 912 ?A=C WP Match?
653 001F9 80 GOYES TblS19 YES!
654 001FB 167 D0=D0+ 8 No. D0 past CON(5) & REL(3).
655 001FE 50A GONC TblS05 (BET) Try next entry.
656 *
657 00201 146 TblS19 C=DAT0 A Read address of PEEK.
658 00204 164 D0=D0+ 5 Point to REL(3).
659 00207 111 A=R1 Fetch addr past string
660 0020A 131 D1=A arg, put in D1.
661 0020D AF0 A=0 W
662 00210 8D00 GOVLNG =TBLJMJ2 (0243D) Jump to REL3D0, put
000
663 *
664 *
665 00217 862 Return ?ST=0 NumArg SYSTEM$(<arg#>)?
666 0021A 80 GOYES Retrn3 No.
667 0021C 137 CD1EX Yes. Save counter
668 0021F 10A R2=C in R2.
669 00222 136 Retrn3 CDOEX Addr of zero nib at tbl end.
670 00225 E6 C=C+1 A Past zero nib.
671 00227 06 RSTK=C Jump back to this address
672 00229 03 RTNCC w/carry clear for processing;
673 continue poll handling.
674 *
675 *----- * For SYSTEM$(<arg#>): return tabled string as function value.
676 *
677 0022B AF1 Sfound B=0 W B(S)=0 and B(A)=0.
678 0022E AE5 B=C B B(1)=#chars-1 in entry.
679 00231 F5 BSR A #chars-1 to B(0).
680 00233 E5 B=B+1 A B(A)=#chars.
681 00235 E5 B=B+1 A #chars in full string.
682 00237 C5 B=B+B A B(A)=length of string.
683 00239 132 ADOEX Addr of table entry to A.
684 0023C 22 P= 2 Read chars from A(A) addr, REV$.
685 0023E 821 XM=0 "Handled".
686 00241 03 RTNCC
687 ****
688 *
689 *
690 00243 AF6 HNDL=1 C=A W Fetch hex value from A.

```

Saturn Assembler SYSTEM LEX file <850429.0820> Mon Apr 29, 1985 8:21 am
Ver. 3.39/Rev. 2306 Keyword execution Page 14

```

691 00246 8F00 GOSBVL =HXDCW (0ECB4) Convert to decimal.
000
692 0024D 8F00 GOSBVL =FLOAT (1B322) Float to 12-digit form in A.
000
693 00254 118 HNDL=0 C=R0 Fetch PC.
694 00257 134 D0=C Restore PC.
695 0025A AF6 C=A W Copy stack value.
696 0025D 860 ?ST=0 Systm$ SYSTEM$ function?
697 00260 80 GOYES Hnd1-1 No, SYSTEM function.
698 00262 B04 A=A+1 P Yes. If string, set carry.
699 00265 590 GONC Hnd1-3 NC=numeric (real or cplx).
700 00268 8D00 Hnd1-1 GOVLNG =FNRTN1 (0F0216) Check AvMem, write to stack.
000
701 *
702 0026F 1CF Hnd1-3 D1=D1- 16 Value to stack (string hdr was
703 00272 1557 DAT1=C W 16 nibs, so real will fit).
704 00276 8D00 GOVLNG =STR$ (18156) Convert to strg, exit to EXPR.
000
705 *
706 0027D A4D HNDL=2 B=B-1 S B(S)#0: "Reverse string."
707 00280 HNDL=3 B(S)=0: "Don't reverse string."
708 00280 130 D0=A Addr of chars to D0.
709 00283 D9 C=B A C(A)=#nibs in string.
710 00285 8F00 GOSBVL =STRHDR (0F09A) Write str hdr, R1=addr of hdr.
000
711 0028C 8F00 GOSBVL =MOVEU0 (1B162) Move chars to string position.
000
712 00293 119 C=R1 Addr of new string header.
713 00296 135 D1=C
714 00299 949 ?B=0 S Reverse string?
715 0029C C2 GOYES Hnd3-5 No.
716 0029E 862 ?ST=0 NumArg SYSTEM$ with numeric arg?
717 002A1 02 GOYES Hnd3-3 No.
718 002A3 17F D1=D1+ 16 Yes, SYSTEM(<arg#>). D1 to
719 002A6 14F C=DAT1 B 1st character, read into C.
720 * C(0)=10xy. x=SYSTEM, y=SYSTEM$.
721 * C(0)=9 (s), A (n) or B (b).
722 002A9 CE C=C-1 A C(0)=8 (s), 9 (n) or A (b).
723 002AB C6 C=C+C A C(0)=0 (s), 2 (n) or 4 (b).
724 002AD FA C=-C A C(0)=0 (s), E (n) or C (b).
725 002AF 8D00 P=C 0
726 002B3 3537 LCASC \ ns\ E602
727 002BB 14D DAT1=C B Replace 1st char with n,s or b.
728 002BE 1CF D1=D1- 16 D1=addr of string header.
729 002C1 8F00 Hnd3-3 GOSBVL =REV$ (1B38E) Reverse (sets P=0).
000
730 002C8 1537 Hnd3-5 A=DAT1 W Read back header.
731 002C C 17F D1=D1+ 16 D1 to top of math stack.
732 002C F 548 GONC HNDL=0 (BET)
733 ****

```

Saturn Assembler SYSTEM LEX file <850429.0820> Mon Apr 29, 1985 8:21 am
Ver. 3.39/Rev. 2306 Poll handlers Page 15

```
734      STITLE Poll handlers
735      ****
736      ** Two poll handlers included: VER$ poll, and pSysFn poll.
737      **
738      ** If you are extending the SYSTEM/SYSTEM$ function, you
739      ** should leave the VER$ poll as is; the VER$ response is to
740      ** be changed only if the SYSTEM/SYSTEM$ code itself changes.
741      **
742      ** You should modify the pSysFn poll handler if you are ex-
743      ** tending the SYSTEM/SYSTEM$ functions to add new valid arg-
744      ** uments. You can add or delete entries from the list below
745      ** as desired, or replace them entirely with your own. All
746      ** code above this point should be copied verbatim for assem-
747      ** bling the SYSTEM/SYSTEM$ execution routines (unless the
748      ** functions are already in memory from another application).
749      **
750      ** See the pSysFn poll interface for details.
751      ****
752      stype EQU 1           string: SYSTEM$ only.
753      ntype EQU 2           numeric: SYSTEM only.
754      btype EQU (stype)+(ntype) both: SYSTEM or SYSTEM$.
755      *
756      *
757 002D2 96D POLLhm ?B#0 B     VER$ poll?
758 002D5 93 GOYES POLLh3    No.
759 002D7 11B VER$P C=R3
760 002DA 135 D1=C
761 002DD 112 A=R2
762 002E0 1CB D1=D1- (VER$en)-(VER$st)-2
763 002E3 137 CD1EX
764 002E6 8B6 ?A>C A
765 002E9 A1 GOYES NtHNDL
766 002EB 10B R3=C
767 002EE 137 CD1EX
768 002F1 3B14 VER$st LCASC \ SYS:A\
A335
9535
02
769 002FF 15DB VER$en DAT1=C (VER$en)-(VER$st)-2
770 00303 FE NtHNDL C=-C-1 A Clear carry.
771 00305 00 RTNSKM "Not Handled".
772
773
774
775 00307 11B PolSYS C=R3 Address of TblScn.
776 0030A 06 RSTK=C
777 0030C 03 RTNCC Jump to TblScn for table scan.
778
779 0030E 3144 POLLh3 LC(2) =pSysFn SYSTEM/SYSTEM$ poll?
780 00312 965 ?B#C B
781 00315 EE GOYES NtHNDL No. "Not Handled".
782 00317 7CEF GOSUB PolSYS Get address of SYSTEM table.
783
784      *----- SYSTEM/SYSTEM$ argument table -----
785
```

Saturn Assembler SYSTEM LEX file <850429.0820> Mon Apr 29, 1985 8:21 am
Ver. 3.39/Rev. 2306 Poll handlers Page 16

```
786 0031B B CON(1) 8+btype Characterization nibble.
787 0031C 4 CON(1) ((end1)-(start1))/2-1 Length=#chars-1.
788 0031D 7594 start1 NIBASC \WIDTH\ Table entry.
4445
84
789 00327 0000 end1 CON(5) =DWIDTH (2F94F) Address of PEEK value.
0
790 0032C 221 REL(3) Swidth Rel offset to processor.
791
792 0032F B CON(1) 8+btype
793 00330 5 CON(1) ((end2)-(start2))/2-1
794 00331 0575 start2 NIBASC \PWIDHT\ 
9444
4584
795 0033D 0000 end2 CON(5) =PWIDHT (2F958)
0
796 00342 C01 REL(3) Spwdth
797
798 00345 9 CON(1) 8+stype
799 00346 7 CON(1) ((end3)-(start3))/2-1
800 00347 0514 start3 NIBASC \PASSWORD\ 
3535
75F4
2544
801 00357 0000 end3 CON(5) =LOCKWD (2F7B2)
0
802 0035C B31 REL(3) Spsswd
803
804 0035F B CON(1) 8+btype
805 00360 5 CON(1) ((end4)-(start4))/2-1
806 00361 34D4 start4 NIBASC \CMDSTK\ 
4435
45B4
807 0036D 0000 end4 CON(5) =MAXCMD (2F976)
0
808 00372 001 REL(3) Scmdst
809
810 00375 B CON(1) 8+btype
811 00376 7 CON(1) ((end5)-(start5))/2-1
812 00377 B454 start5 NIBASC \KEYCOUNT\ 
9534
F455
E445
813 00387 0000 end5 CON(5) =KEYPTR (2F443)
0
814 0038C FD0 REL(3) Skeybf
815
816 0038F B CON(1) 8+btype
817 00390 4 CON(1) ((end6)-(start6))/2-1
818 00391 34C4 start6 NIBASC \CLOCK\ 
F434
B4
819 0039B 0000 end6 CON(5) =CSPEED (2F977)
0
820 003A0 BD0 REL(3) Sclock
```

Saturn Assembler SYSTEM LEX file <850429.0820> Mon Apr 29, 1985 8:21 am
Ver. 3.39/Rev. 2306 Poll handlers

```

821      *
822 003A3 9      CON(1) 8+stype
823 003A4 6      CON(1) ((end7)-(start7))/2-1
824 003A5 54E4  start7 NIBASC \ENDLINE\
44C4
94E4
54
825 003B3 0000  end7  CON(5) =EOLLEN (2F95A)
0
826 003B8 CC0    REL(3) Sendln
827      *
828 003BB 9      CON(1) 8+stype
829 003BC 7      CON(1) ((end8)-(start8))/2-1
830 003BD 4494  start8 NIBASC \DISPFORM\
3505
64F4
25D4
831 003CD 0000  end8  CON(5) =DSPFM (2F6DC)
0
832 003D2 BD0    REL(3) Sdisp
833      *
834 003D5 B      CON(1) 8+btype
835 003D6 7      CON(1) ((end9)-(start9))/2-1
836 003D7 34F4  start9 NIBASC \CONTRAST\
E445
2514
3545
837 003E7 0000  end9  CON(5) =DCONTR (2E3FE)
0
838 003EC F70    REL(3) Sctrst
839      *
840
841 003EF B      CON(1) 8+btype
842 003F0 5      CON(1) ((end10)-(strt10))/2-1
843 003F1 C444  strt10 NIBASC \LDELAY\
54C4
1495
844 003FD 0000  end10 CON(5) =DELAYT (2F948)
0
845 00402 C10    REL(3) Sldely
846      *
847 00405 B      CON(1) 8+btype
848 00406 5      CON(1) ((end11)-(strt11))/2-1
849 00407 3444  strt11 NIBASC \CDELAY\
54C4
1495
850 00413 0000  end11 CON(5) =SCROLLT (2F946)
0
851 00418 600    REL(3) Scdely
852      *
853 0041B 0      CON(1) 0      End of table.
854 0041C 00     RTNSXM      Not handled by this LEX file.
855      *
856      *----- end of SYSTEM/SYSTEM$ argument table -----
```

Saturn Assembler SYSTEM LEX file <850429.0820> Mon Apr 29, 1985 8:21 am
Ver. 3.39/Rev. 2306 Poll handlers

```

857      EJECT
858      ****
859      ** The following processing routines are part of the pSysFn
860      ** poll handler. The above SYSTEM/SYSTEM$ table includes
861      ** relative offsets to the individual processors. See the
862      ** pSysFn poll interface for more details, under 'Part II'.
863      ****
864      *
865 0041E Scdely
866 0041E 14A  Sldely A=DATA0 B      Delay rate in 1/32s.
867 00421 968  ?A=0 B
868 00424 C3   GOYES Swd-03      If zero, skip math (faster, and
869 00426 B64   A=A+1 B      otherwise gives denormalized 0).
870 00429 4C2   GOC wd=inf      Inf?
871 0042C CC   A=A-1 A      Yes.
872 0042E 8F00  GOSBVL =HDFLT (1B31B) Float.
000
873 00435 AF2   C=0 W
874 00438 2D   P= 13
875 0043A 3323  LCHEX 1032      Decimal 32.
01
876 00440 8F00  GOSBVL =DV2-12 (0C4A8) Divide delay rate by 32.
000
877 00447 AD4   A=B M      Truncate to 12-digit form.
878 0044A 6510  GOTO Swd-03      Exit.
879      *-----
880 0044E Swidth
881 0044E 14A  Spwdth A=DATA0 B      WIDTH setting.
882 00451 96C  ?A#0 B      '00' means Inf.
883 00454 E0   GOYES Swd-05
884 00456 8F00  wd=inf GOSBVL =HUGE (0B75D) Generate "Inf".
000
885 0045D AFA   Swd-01 A=C W      Leave function value in A.
886 00460 2F   Swd-03 P= 15      Make P=0:"Dec value in A."
887 00462 0C   Swd-05 P=P+1      P=1:"Hex value in A."
888 00464 04   Swd-07 SETHEX
889 00466 821   XM=0
890 00469 03   RINCC
891      *-----
892 0046B Sctrst
893 0046B 15A0  Skeybf A=DATA0 1
894 0046F 52F   GONC Swd-05      (BET) Hex value in A.
895      *-----
896 00472 15A0  Scmdst A=DATA0 1
897 00476 E4   A=A+1 A
898 00478 59E   GONC Swd-05      (BET) Hex value in A.
899      *-----
900 0047B 142   Sclock A=DATA0 A
901 0047E BF0   ASL W
902 00481 50E   GONC Swd-05      Multiply by 16.
903      *-----
904 00484 15A0  Sendln A=DATA0 1
905 00488 160   'DO=D0+ 1      (BET) Hex value in A.
906 0048B AF8   Send13 B=A W      #nibs in ENDLN string.
907 0048E 132   ADOEX      Addr of ENDLN string.
908      *-----
```

Saturn Assembler SYSTEM LEX file <850429.0820> Mon Apr 29, 1985 8:21 am
Ver. 3.39/Rev. 2306 Poll handlers Page 19

```

908 00491 22      P=    2          "Read chars from A(A) addr,
909 00493 60DF    GOTO  Swd-07    and reverse."
910 *-----*
911 00497 1567    Spsswd C=DATO W  Read entire password.
912 0049B 97A     Spssw1 ?C=0 W   End of password?
913 0049E DE      GOYES Send13  Yes. Write out chars.
914 004A0 E4      A=A+1 A      Count #nibs.
915 004A2 E4      A=A+1 A
916 004A4 BF6     CSR   W
917 004A7 BF6     CSR   W
918 004AA 50F     GONC  Spssw1  (BET)
919 *-----*
920 004AD 14A     Sdisp  A=DATO B  Lower two bits of A(0): 0=STD,
921 004B0 3130    LCHEX 03       1=FIX, 2=SCI, 3=ENG.
922 004B4 D5      B=C   A       Mask for lower bits.
923 004B6 0EF4    B=A&B A      B(A)=0,1,2 or 3.
924 004BA 7810    GOSUB Sdisp3  Get addr of ASCII.
925 004BE 4445    NIBASC \DTSXIF\
3585
9464
926 004CA 9434    NIBASC \ICSGNE\
3574
E454
927 004D6 07     Sdisp3 C=RSTK  Addr of ASCII.
928 004D8 C5      B=B+B A      2x
929 004DA C9      C=B+C A      +2x
930 004DC C5      B=B+B A      4x
931 004DE C9      C=B+C A      +6x
932 004E0 134    D0=C       Address of ASCII dispform.
933 004E3 15E5    C=DATO 6      Read display setting.
934 004E7 1C5     D1=D1- 6      Back up stack pointer.
935 004EA 15D5    DAT1=C 6      Write out.
936 004EE 969     ?B=0 B      STD?
937 004F1 02      GOYES Sdisp5  Yes.
938 004F3 1C5     D1=D1- 6      In case two digits.
939 004F6 F4      ASR   A      #digits to A(P).
940 004F8 35A3    LCHEX 20313A Blank, ASCII "1x".
1302
941 00500 B0E     C=A-C P      If #dig<=9, sets carry.
942 00503 15D5    DAT1-C 6      Write out blank, 2 digits.
943 00507 590     GONC  Sdisp5  NC= #digs>10.
944 0050A 171     D1=D1+ 2      Correct for one digit.
945 0050D 1590    DAT1=A 1      Write out one digit.
946 00511 133    Sdisp5 AD1EX  A(A)=addr of string. A(15-5)=0.
947 00514 AF6     C=A   W      Addr of string to C. C(15-5)=0.
948 00517 121     AR1EX
949 0051A EE      C=A-C A      R1=new top of stack, A=old.
950 0051C F2      CSL   A      #nibs in string (<= 12 dec.)
951 0051E F2      CSL   A      String length to C(6-2).
952 00520 30F     LCHEX F      (A fld OK, since len<=12).
953 00523 693F    GOTO  Swd-01 String header in C.
954 *-----*
955 00527        FILEND END

```

Saturn Assembler SYSTEM LEX file <850429.0820> Mon Apr 29, 1985 8:21 am
Ver. 3.39/Rev. 2306 Symbol Table Page 20

	CONVUC	Ext	-	591
	CSPED	Ext	-	819
	ChrCnt	Abs	3 #00003	511
	DO=AVS	Ext	-	573
	DCONTR	Ext	-	837
	DELAYT	Ext	-	844
	DSPFMT	Ext	-	831
	DV2-12	Ext	-	876
	DWUDTH	Ext	-	789
	EOLLEN	Ext	-	825
	FILEND	Rel	1319 #00527	955
	FLOAT	Ext	-	692
	FLTDH	Ext	-	538
	FNRTN1	Ext	-	700
	FPOLL	Ext	-	547
	HDFLT	Ext	-	872
	HNDL=0	Rel	596 #00254	693
	HNDL=1	Rel	579 #00243	690
	HNDL=2	Rel	637 #0027D	706
	HNDL=3	Rel	640 #00280	707
	HUGE	Ext	-	556
	HXDCW	Ext	-	884
	Hnd1-1	Rel	616 #00268	700
	Hnd1-3	Rel	623 #0026F	702
	Hnd3-3	Rel	705 #002C1	729
	Hnd3-5	Rel	712 #002C8	730
	KEYPTR	Ext	-	717
	LOCKWD	Ext	-	813
	MAXCMD	Ext	-	801
	MFERR	Ext	-	807
	MOVEU0	Ext	-	563
	Memerr	Rel	272 #00110	565
	NotFnd	Rel	250 #000FA	558
	NtHNDL	Rel	771 #00303	539
	NumArg	Abs	2 #00002	770
	POLLh3	Rel	782 #0030E	765
	POLLhn	Rel	722 #002D2	758
	POP1R	Ext	-	627
	POPMT	Ext	-	665
	PWIDHT	Ext	-	716
	PolSYS	Rel	775 #00307	541
	REV\$	Ext	-	549
	Retrn3	Rel	546 #00222	729
	Return	Rel	535 #00217	669
	SCROLLT	Ext	-	626
	STR\$	Ext	-	850
	STRHDR	Ext	-	704
	SYS01	Rel	134 #00086	710
	Scdely	Rel	1054 #0041E	523
	Sclock	Rel	1147 #0047B	865
	Scmdst	Rel	1138 #00472	851
	Sctrst	Rel	1131 #0046B	900
	Sdisp	Rel	1197 #004AD	820
	Sdisp3	Rel	1238 #004D6	832
	Sdisp5	Rel	1297 #00511	924
				946
				937
				943

Saturn Assembler SYSTEM LEX file <850429.0820> Mon Apr 29, 1985 8:21 am
Ver. 3.39/Rev. 2306 Symbol Table Page 21

Sendl3	Rel	1163	#0048B	-	906	913				
Sendln	Rel	1156	#00484	-	904	826				
Sfounds	Rel	555	#0022B	-	677	630				
Skeybf	Rel	1131	#0046B	-	893	814				
Sldely	Rel	1054	#0041E	-	866	845				
Spssw1	Rel	1179	#0049B	-	912	918				
Spsswd	Rel	1175	#00497	-	911	802				
Spwdth	Rel	1102	#0044E	-	881	796				
Swd-01	Rel	1117	#0045D	-	885	953				
Swd-03	Rel	1120	#00460	-	886	868	878			
Swd-05	Rel	1122	#00462	-	887	883	894	898	901	
Swd-07	Rel	1124	#00464	-	888	909				
Swidth	Rel	1102	#0044E	-	880	790				
Sxx03	Rel	279	#00117	-	568	535				
Sxx05	Rel	311	#00137	-	578	596				
Sxx07	Rel	315	#0013B	-	579	585				
Sxx09	Rel	347	#0015B	-	591	587				
Sxx19	Rel	369	#00171	-	598	580				
Sxx21	Rel	377	#00179	-	601	589				
Sxx23	Rel	389	#00185	-	605	543				
SysP11	Rel	225	#000E1	-	550	561				
SySpol	Rel	208	#000DO	-	545	605				
System	Abs	1	#00001	-	509	522				
Systm\$	Abs	0	#00000	-	508	517	696			
TBLJM2	Ext			-	662					
TBLJMP	Ext			-	552					
Tbls03	Rel	405	#00195	-	619	616				
Tbls05	Rel	415	#0019F	-	623	655				
Tbls07	Rel	436	#001B4	-	631	628				
Tbls09	Rel	439	#001B7	-	632	644				
Tbls11	Rel	457	#001C9	-	638	636				
Tbls15	Rel	489	#001E9	-	648	640				
Tbls17	Rel	493	#001ED	-	649	633				
Tbls19	Rel	513	#00201	-	657	653				
TblScn	Rel	393	#00189	-	613					
TxTbSt	Rel	80	#00050	-	31	17				
VER\$P	Rel	727	#002D7	-	759					
VER\$en	Rel	767	#002FF	-	769	769	762			
VER\$st	Rel	753	#002F1	-	768	769	762			
XSys\$	Rel	118	#00076	-	516	23				
XSys\$tm	Rel	129	#00081	-	521	27				
btype	Abs	3	#00003	-	754	786	792	804	811	
					847					
eIVARG	Ext			-	562					
eMEM	Ext			-	565					
end1	Rel	807	#00327	-	789	787				
end10	Rel	1021	#003FD	-	844	842				
end11	Rel	1043	#00413	-	850	848				
end2	Rel	829	#0033D	-	795	793				
end3	Rel	855	#00357	-	801	799				
end4	Rel	877	#0036D	-	807	805				
end5	Rel	903	#00387	-	813	811				
end6	Rel	923	#0039B	-	819	817				
end7	Rel	947	#003B3	-	825	823				
end8	Rel	973	#003CD	-	831	829				

Saturn Assembler SYSTEM LEX file <850429.0820> Mon Apr 29, 1985 8:21 am
Ver. 3.39/Rev. 2306 Symbol Table Page 22

end9	Rel	999	#003E7	-	837	835			
errarg	Rel	261	#00105	-	562	599			
error	Rel	265	#00109	-	563	566			
fLEX	Ext			-	5				
ntype	Abs	2	#00002	-	753	754			
pSysFn	Abs	68	#00044	-	512	548	779		
start1	Rel	797	#0031D	-	788	787			
start2	Rel	817	#00331	-	794	793			
start3	Rel	839	#00347	-	800	799			
start4	Rel	865	#00361	-	806	805			
start5	Rel	887	#00377	-	812	811			
start6	Rel	913	#00391	-	818	817			
start7	Rel	933	#003A5	-	824	823			
start8	Rel	957	#003BD	-	830	829			
start9	Rel	983	#003D7	-	836	835			
strt10	Rel	1009	#003F1	-	843	842			
strt11	Rel	1031	#00407	-	849	848			
stype	Abs	1	#00001	-	752	754	798	822	820
wd=inf	Rel	1110	#00456	-	884	870			