```
498 C60B 7B                 MOV   A,E       Get digit count
499 C60C 32F100             STA   :00F1     Store it in buffer
500 C60F C9                 RET
501                 *
502                 * DATA:
503                 *
504 C610 00         I10     DATA  :00       INT (10)
505 C611 00                 DATA  :00
506 C612 00                 DATA  :00
507 C613 0A                 DATA  :0A
508                 *
509                 *
510                 *
511 C614                    END
```

```
****************************
* S Y M B O L   T A B L E *
****************************
```

FP0	C45E	FP1	C462	FP2	C466	FP3	C46A
FP4	C46E	FP5	C472	FP6	C476	FP7	C47A
FP8	C47E	FP9	C482	I10	C610	IBC	C5B2
ICB	C573	LC100	C502	LC101	C513	LC102	C51A
LC103	C525	LC104	C531	LC105	C53C	LC106	C54B
LC107	C554	LC108	C55F	LC109	C56F	LC110	C57A
LC111	C590	LC112	C598	LC113	C5A5	LC114	C5A7
LC115	C5BB	LC116	C5E0	LC117	C5F0	LC118	C5FA
LC119	C606	LC234	C437	LC238	C45A	LC91	C498
LC92	C4A3	LC93	C4AF	LC94	C4C3	LC95	C4C7
LC96	C4D5	LC97	C4D8	LC98	C4E1	LC99	C4FF
PRTY	C486						


```
002                         ORG   :C614
003                 *
004                 *
005                 *
006                 ****************************
007                 * INPUT HEX NUMBER TO MACC *
008                 ****************************
009                 *
010                 * Reads a sequence of hex digits and converts
011                 * them into MACC.
012                 *
013                 * Entry: C points to input.
014                 * Exit:  CY=1: There was a digit.
015                 *        CY=0: No digit.
016                 *        C points to next input.
017                 *        ABDEHL preserved.
018                 *
019 C614 37         HCB     STC
020 C615 F5                 PUSH  PSW
021 C616 D5                 PUSH  D
022 C617 E5                 PUSH  H
023 C618 CD98C5             CALL  :C598     Clear MACC and 00E3-E6
024 C61B CD73C0     LC120   CALL  :C073     Get digit from line
025 C61E D630               SUI   :30       )
026 C620 DA90C5             JC    :C590     )
027 C623 FE0A               CPI   :0A       ) Check if hex number
028 C625 DA34C6             JC    :C634     )
029 C628 D607               SUI   :07       ) Abort via C590 if not
030 C62A FE0A               CPI   :0A       )
031 C62C DA90C5             JC    :C590     )
032 C62F FE10               CPI   :10       )
033 C631 D290C5             JNC   :C590     )
034 C634 213DC6     LC121   LXI   H,:C63D   Addr INT(4)
035 C637 CD41C6             CALL  :C641     Insert digit at low end
036                                         MACC
037 C63A C31BC6             JMP   :C61B     Get next digit
038                 *
039                 * DATA:
040                 *
041 C63D 00         I4      DATA  :00       INT (4)
042 C63E 00                 DATA  :00
043 C63F 00                 DATA  :00
044 C640 04                 DATA  :04
045                 *
046                 * ENTER HEX DIGIT AT LOW END MACC:
047                 *
048                 * Entry: HL points to a 4-byte number.
049                 *        A  contains a digit.
050                 * Exit:  HL = 00E3.
051                 *        C is incremented, D decremented.
052                 *        ABE preserved.
053                 *
054 C641 F5         LC122   PUSH  PSW
055 C642 C5                 PUSH  B
056 C643 D5                 PUSH  D
057 C644 E7                 RST   4         Copy MACC to reg A,B,C,D
058 C645 15                 DATA  :15
059 C646 E6F0               ANI   :F0       Check if value too high
060 C648 C44BC0             CNZ   :C04B     Then overflow error
061 C64B D1                 POP   D
062 C64C C1                 POP   B
063 C64D F1                 POP   PSW
```