

```

0368 F8 LDI
69 04
6A AE PLO RE ;Set loop counter = 04
6B 0F LDN RF ;Get character bit pattern (2 rows)
6C FA ANI ;AND with F0 for MSB's
6D F0
6E BE PHI RE ;Store in RE.1 to pass to sub
6F D4 SEP R4
0370 03 ;Call Display Bit Row @ R7
71 88
72 4F LDA RF ;Get same bit pattern
73 FE SHL ;Shift left for LSB's
74 FE SHL ; " " "
75 FE SHL ; " " "
76 FE SHL ; " " "
77 BE PHI RE ;Store in RE.1 to pass to sub
78 D4 SEP R4
79 03 ;Call Display Bit Row @ R7
7A 88
7B 2E DEC RE ;Loop count - 01
7C 8E GLO RE
7D 3A BNZ ;Loop 4 times total to display eight bit rows
7E 6B
7F 87 GLO R7
0380 FF SMI ;Subtract 40 hex from cursor address
81 40
82 A7 PLO R7 ;To reset to top of character rows
83 97 GHI R7
84 7F SMBI ;(This may be shortened by 3 bytes by using
85 00 ;A separate register for display)
86 B7 PHI R7
87 D5 SEP R5 ;Return

```

DISPLAY BIT ROW-DOES ONE OF 8 BIT ROW PATTERNS FOR A CHARACTER

```

0388 8A GLO RA ;Data displacement pointer
89 F6 SHR ;Shift to test even/odd
8A 3B BNF ;Branch if even (DF=0)
8B 92 ;To skip the shifting (display bits on left)
8C 9E GHI RE ;Get unpacked bits in RE.1 (RA is odd)
8D F6 SHR ;Shift right x 4 (display bits on right)
8E F6 SHR
8F F6 SHR
0390 F6 SHR
91 BE PHI RE ;Store processed bit row in RE.1
92 E7 SEX 7 ;X = 7 (R7 is display cursor)
93 9E GHI RE ;Get the processed bit row
94 F3 XOR ;Exclusive OR with what is already displayed
95 57 STR R7 ;Put in display @ R7

```