```
024B   3F                   ;If not continue
 4C    F8    LDI            ;Else load RC
 4D    09
 4E    BC    PHI   RC       ;With error message (???)
 4F    F8    LDI
0250   93                   ;Address
 51    AC    PLO   RC
 52    D5    SEP   R5       ;And return
```

SPECIAL CASES DECODING

```
0253   E2    SEX   2        ;X = 2
 54    0A    LDN   RA       ;Get the instruction for disassembly
 55    32    BZ             ;If = 00 (IDL) Branch to exit @ 028F
 56    8F
 57    FA    ANI
 58    F0                   ;Strip second digit to test for "6N" In/Out bytes
 59    FB    XRI
 5A    60
 5B    3A    BNZ            ;If not a 6N type, branch to next test
 5C    73
 5D    0A    LDN   RA       ;Get the instruction
 5E    FB    XRI
 5F    60                   ;If 6N type is 60, IRX, branch
0260   32    BZ
 61    8F                   ;To exit @ 028F
 62    0A    LDN   RA
 63    FB    XRI
 64    68                   ;If 6N type is an illegal code, (68)
 65    32    BZ
 66    8F                   ;Branch to exit @ 028F (Error will be output to text)
 67    0A    LDN   RA
 68    FD    SDI            ;Subtract -- 67 - instruction (If > 67, then
 69    67                   ;Result negative, if = or < 67, then result positive)
 6A    33    BPZ            ;Branch if positive to set X of NX = F
 6B    8A                   ;(6N type is an out instruction)
 6C    0A    LDN   RA       ;Get instruction (now determined to be an "INP")
 6D    FA    ANI
 6E    F0                   ;Strip the second digit
 6F    F9    ORI
0270   01                   ;OR with 01 to put a 1 in second digit
 71    B7    PHI   R7       ;Store in R7.1 for look up table
 72    D5    SEP   R5       ;Return
 73    0A    LDN   RA       ;Get instruction for disassembly
 74    FA    ANI
 75    F0                   ;Strip the second digit
 76    52    STR   R2       ;Push for comparison tests
 77    FB    XRI
 78    30                   ;If = 30 (3N type)
```