

Program

The bootstrap loader contains two locations which are modified by reading the input: the first is a [pointer](#) to the [buffer](#) in which to place data being read in; the second is a [branch](#) back to the beginning of the data input [loop](#).

The program starts with the pointer set to itself, so the blank leader at the start of the data for the bootstrap loader (originally contained on [paper tapes](#)) starts with a special leader code which causes this to be left un-changed.

The buffer pointer is then modified to point it slightly before the bootstrap loader in memory, so at the end of the load the newly loaded code then progressively overlays the first several [instructions](#) of the bootstrap loader itself, finally patching the buffer pointer to point at the branch instruction, which is then modified to jump to the newly loaded code.

Code

This is the actual bootstrap program:

```

                LOAD = 0x7400                                ; Buffer address
0x7744 016701 BEGIN: MOV     DEVICE, R1                    ; Get Device CSR
0x7746 000026
0x7750 012702 LOOP:  MOV     (PC)+, R2                    ; Get buffer offset
0x7752 000352 OFFSET: .-LOAD
0x7754 005211          INC     @R1                        ; Turn on reader
0x7756 105711 READY: TSTB   @R1                          ; Done?
0x7760 100376          BPL     READY
0x7762 116162          MOVB   2(R1), LOAD(R2) ; Transfer
0x7764 000002
0x7766 0x7400
0x7770 005267          INC     OFFSET                    ; Bump buffer offset
0x7772 177756
0x7774 000765          BR     LOOP
0x7776 yyyyyy DEVICE: yyyyyy                            ; Input device CSR address
```

Bootstrap Loader format

The format of bootstrap loader input is as follows:

- Leader (all bytes 0351)
- Load offset (1 byte)
- Program to be loaded (up to 0344 bytes)
- Copy of bootstrap loader (first 6 bytes)
- Offset to branch (1 byte)
- Branch modification (1 byte)