


```

1          title      Xtract field from and in
2          name      ('XTRINS')
3          ;
4          ; This is intended for use with
5          ;
6          .list      ; re-enable listing
7          ;
8          ;
9          ;++ *****
10         ;
11         ; TO USE THIS ROUTINE:
12         ;
13         ;          M80 XTRINSLB=XTRINSLB
14         ;          L80 XTRINSLB,XTRINSLB/N/
15         ;          REN BIOS.CRL=BIOS.COM
16         ;
17         ;          XTRINSLB.CRL WILL BE A '
18         ;          WHICH CAN BE REQUESTED A
19         ;          INTO YOUR LIBRARY WITH C
20         ;
21         ;-- *****
22         ;
23         03F7          MAGIC equ      3F7H          ; 'C' PAR
24         ;
25         0000'        aseq
26         ;
27         ;          org      100H
28         ;          .phase  0
29         0000 49 4E 53 45          ZERO:  dc      'INSERT'          ; Name o
30         0004 52 D4
31         0006 0205          dw      INSHEAD          ; Locati
32         0008 58 54 52 41          dc      'XTRACT'          ; Name o
33         000C 43 D4
34         000E 0284          dw      XTRHEAD          ; Locati
35         ;
36         0010 80          db      80H          ;END OF
37         0011 02C8          dw      FINIS - 100H          ;POINTER
38         0013          ds      (512 - ($-ZERO))          ;PAD RES
39         0200          ds      5          ;RESERVE
40         ;
41         .dephase
42         ;
43         page

```

```

44      0305
45
46      ;
47      ;++ *****
48      ;$$ INSERT      -- Insert bit fi
49      ;
50      ;      Bits are numbered 1..N
51      ;      Width may be 1..16
52      ;
53      ;      Storage format is:
54      ;
55      ;          msb      lsb
56      ;
57      ;          8        1
58      ;          16       9
59      ;          .        .
60      ;          .        .
61      ;          N        N-8
62      ;
63      ; insert(array,data,start,width)
64      ; char *array;
65      ; unsigned data;
66      ; char start,width;
67      ; {
68      ;     .
69      ;     .
70      ;     return 0;
71      ; }
72      ;
73      ;
74      ; Warning:      array must be at
75      ;                (start+width)/8
76      ;
77      ;-- *****
78      ;
79      .phase $-100H
80      ;
81      INSHEAD:
82      0205      00      db      0      ; No external fu
83      0206      007A    dw      INSTOP-INSERT ; length
84      ;
85      .dephase
86      ;
87      ; Body of function INSERT
88      ;

```

```

89          .phase 0
90          ;
91      0000      2A 03F7          INSERT:  LHLD      MAGIC      ; HL = .(byte ar
92          LDED      MAGIC+2 ; DE = data to b
93      0003      ED 5B          db      0EDH,5BH
94      0005      03F9          dw      MAGIC+2
95          LBCD      MAGIC+4 ; C = starting
96      0007      ED 4B          db      0EDH,4BH
97      0009      03FB          dw      MAGIC+4
98      000B      3A 03FD          LDA      MAGIC+6
99      000E      47          MOV      B,A      ; B = field wid
100     000F      0D          DCR      C      ; bit no. 1..256
101     0010      E5          PUSH     H      ; save pointer t
102     0011      21 0001          LXI     H,1     ; calculate data
103     0014      29          MSLUP:  DAD      H
104          DJNZ     MSLUP
105     0015      10 FD          db      10H,MSLUP-$-1
106     0017      2B          DCX      H      ; HL = 2**width
107     0018      7B          MOV      A,E
108     0019      A5          ANA     L      ; mask the data
109     001A      5F          MOV      E,A
110     001B      7A          MOV      A,D
111     001C      A4          ANA     H
112     001D      57          MOV      D,A
113     001E      7C          MOV      A,H      ; invert the mas
114     001F      2F          CMA
115     0020      67          MOV      H,A
116     0021      7D          MOV      A,L
117     0022      2F          CMA
118     0023      6F          MOV      L,A
119     0024      E3          XTHL           ; HL = .(array),
120     0025      79          MOV      A,C      ; calc byte offs
121     0026      E6 F8          ANI     not 7
122     0028      0F          RRC
123     0029      0F          RRC
124     002A      0F          RRC
125     002B      85          ADD     L
126     002C      6F          MOV      L,A
127          JRNC    ADOK
128     002D      30 01          db      30H,ADOK-$-1
129     002F      24          INR     H      ; HL = .(three b
130     0030      79          ADOK:  MOV      A,C      ; calc bit in by
131     0031      E6 07          ANI     7
132     0033      3C          INR     A      ; 0..7 -> 1..8
133     0034      4F          MOV      C,A      ; save for right
134     0035      47          MOV      B,A      ; again for left

```

```

135      0036      EB          XCHG          ; DE = .(bytes o
136      0037      E3          XTHL          ; stack = data to
137      0038      E5          PUSH           H          ; stack = invert
138      0039      13          INX            D          ; get 3 byte of
139      003A      13          INX            D
140      003B      1A          LDAX           D
141      003C      67          MOV            H,A
142      003D      1B          DCX            D
143      003E      1A          LDAX           D
144      003F      6F          MOV            L,A
145      0040      1B          DCX            D
146      0041      1A          LDAX           D
147      0042      B7          RJLUP:      ORA            A
148      0043      0D          DCR            C
149          JRZ            RJDUN      ; Is right justi
150      0044      28 0B      db            28H,RJDUN-$-1
151          RARR           H          ; No
152      0046      CB 1C      db            0CBH, 18H +H
153          RARR           L
154      0048      CB 1D      db            0CBH, 18H +L
155      004A      1F          RAR
156          JRNC           RJLUP
157      004B      30 F5      db            30H,RJLUP-$-1
158          SETB           7,H
159      004D      CB FC      db            0CBH,7*8+H+0C0H
160          JMPR           RJLUP
161      004F      18 F1      db            18H,RJLUP-$-1
162          ;
163      0051      EB          RJDUN:      XCHG          ; HL = .(bytes o
164      0052      E3          XTHL          ; HL = inverted
165      0053      A5          ANA            L
166      0054      6F          MOV            L,A
167      0055      7B          MOV            A,E
168      0056      A4          ANA            H
169      0057      5F          MOV            E,A
170      0058      7D          MOV            A,L      ; D'E'A = rj fie
171      0059      E1          POP            H
172      005A      E3          XTHL          ; HL = data to b
173      005B      B5          ORA            L          ; or in the data
174      005C      6F          MOV            L,A
175      005D      7B          MOV            A,E
176      005E      B4          ORA            H
177      005F      5F          MOV            E,A
178      0060      7D          MOV            A,L
179          ;B = shift count
180      0061      B7          LJLUP:      ORA            A

```

```

181      0062      05                DCR      B
182                JRZ      LJDUN      ; Is left justif
183      0063      28 0B            db      28H,LJDUN-$-1
184      0065      17                RAL
185                RALR     E
186      0066      CB 13            db      0CBH, 10H+E
187                RALR     D
188      0068      CB 12            db      0CBH, 10H+D
189                JRNC    LJLUP
190      006A      30 F5            db      30H,LJLUP-$-1
191                SETB    0,A
192      006C      CB C7            db      0CBH,0*8+A+0C0H
193                JMPR    LJLUP
194      006E      18 F1            db      18H,LJLUP-$-1
195                ;
196      0070      E1                LJDUN:  POP      H          ; H1 = .(bytes o
197      0071      77                MOV      M,A
198      0072      23                INX      H
199      0073      73                MOV      M,E
200      0074      23                INX      H
201      0075      72                MOV      M,D          ; modified bytes
202      0076      21 0000          LXI      H,0          ; Function value
203      0079      C9                RET
204                INSTOP:
205      007A      0000            dw      0          ; No non-intrins
206                ;
207                .dephase
208                ;
209                page

```

```

210      0384
211
212      ;
213      ;++ *****
214      ;$$ EXTRACT -- Extract a bit fie
215      ;           -- multi-byte buffer
216      ;
217      ;           Bits are numbered 1..N
218      ;           Width may be 1..16
219      ;
220      ;           Storage format is presum
221      ;
222      ;           msb      lsb
223      ;
224      ;           8        1
225      ;           16       9
226      ;           .        .
227      ;           .        .
228      ;           N        N-8
229      ;
230      ;
231      ; xtract(array,start,width)
232      ;     char *array;
233      ;     char start,width;
234      ;     {
235      ;     .
236      ;     .
237      ;     return <extracted value>
238      ;     }
239      ;
240      ;
241      ;-- *****
242      ;
243      .phase  $-100H
244      ;
245      XTRHEAD:
246      0284      00      db      0
247      0285      003F   dw      XTRTOP-XTRACT
248      ;
249      .dephase
250      ;
251      .phase  0
252      ;
253      0000      2A 03F7 XTRACT: LHLD  MAGIC   ; HL =.(byte arr
254      ;          LBCD  MAGIC+2 ; C = starting

```

```

255      0003      ED 4B                db      0EDH,4BH
256      0005      03F9                dw      MAGIC+2
257      0007      3A 03FB             LDA      MAGIC+4
258      000A      47                   MOV      B,A      ; B = field wid
259      000B      0D                   DCR      C      ; bit no. 1..N
260      000C      79                   MOV      A,C
261      000D      E6 F8               ANI      not 7
262      000F      0F                   RRC
263      0010      0F                   RRC
264      0011      0F                   RRC
265      0012      5F                   MOV      E,A
266      0013      16 00               MVI      D,0      ; DE = byte offs
267      0015      19                   DAD     D      ; HL = .(bytes c
268      0016      79                   MOV      A,C
269      0017      E6 07               ANI      7
270      0019      4F                   MOV      C,A      ; C = bit number
271      001A      0C                   INR      C      ; 0..7 -> 1..8
272      001B      EB                   XCHG    ; DE = .(bytes o
273      001C      13                   INX     D
274      001D      13                   INX     D      ; Could be sprea
275      001E      1A                   LDAX    D      ; get them in H'
276      001F      67                   MOV      H,A
277      0020      1B                   DCX     D
278      0021      1A                   LDAX    D
279      0022      6F                   MOV      L,A
280      0023      1B                   DCX     D
281      0024      1A                   LDAX    D
282      0025      0D                   LOOP:   DCR      C      ; Right justify
283                                     JRZ     JUSTDN ; Is right justifi
284      0026      28 07                db      28H,JUSTDN-$-1
285                                     SRAR    H      ; No
286      0028      CB 2C                db      0CBH, 28H+H
287                                     RARR    L
288      002A      CB 1D                db      0CBH, 18H +L
289      002C      1F                   RAR
290                                     JMPR    LOOP
291      002D      18 F6                db      18H,LOOP-$-1
292                                     ;
293      002F      55                   JUSTDN: MOV     D,L
294      0030      5F                   MOV     E,A      ; max 16 bit fie
295      0031      21 0001             LXI    H,1      ; Calculate the
296      0034      29                   MSKLUP: DAD     H
297                                     DJNZ   MSKLUP
298      0035      10 FD                db      10H,MSKLUP-$-1
299      0037      2B                   DCX     H      ; HL = 2**width
300      0038      7A                   MOV     A,D

```



```
301      0039      A4              ANA      H
302      003A      67              MOV      H,A
303      003B      7B              MOV      A,E
304      003C      A5              ANA      L
305      003D      6F              MOV      L,A      ; HL = extracted
306      003E      C9              RET
307
308
309      003F      0000            XTRTOP:  dw      0
310
311
312
313
314      FINIS:    end      ZERO
```

Macros:

@CHK	BIT	DJNZ	JMPR	JRC	JRNC	JRNZ	JRZ
LBCD	LDED	RALR	RARR	RES	RLCR	RRCR	SETB
SLAR	SRAR	SRLR					

Symbols:

ADOK	0030	FINIS	03C8	INSERT	0000	INSHEA	0205
INSTOP	007A	JUSTDN	002F	LJDUN	0070	LJLUP	0061
LOOP	0025	MAGIC	03F7	MSKLUP	0034	MSLUP	0014
RJDUN	0051	RJLUP	0042	XTRACT	0000	XTRHEA	0284
XTRTOP	003F	ZERO	0000				

No Fatal error(s)

ADOK	128	130#						
DJNZ	104	297						
FINIS	37	314#						
INSERT	83	91#						
INSHEA	31	81#						
INSTOP	83	204#						
JMPR	160	193	290					
JRNC	127	156	189					
JRZ	149	182	283					
JUSTDN	284	293#						
LBCD	95	254						
LDED	92							
LJDUN	183	196#						
LJLUP	180#	190	194					
LOOP	282#	291						
MAGIC	23#	91	94	97	98	253	256	257
MSKLUP	296#	298						
MSLUP	103#	105						
RALR	185	187						
RARR	151	153	287					
RJDUN	150	163#						
RJLUP	147#	157	161					
SETB	158	191						
SRAR	285							
XTRACT	247	253#						
XTRHEA	34	245#						
XTRTOP	247	308#						
ZERO	29#	38	314					

→TOP	83	204#	
JMPR	160	193	290
JRNC	127	156	189
JRZ	149	182	283
JUSTDN	284	293#	
LBCD	95	2	