

```

1 ; Binärzähler
2
3     device 16f84
4
5 ; \***** labels *****\
6
7 ; Statusregister
8 status equ 3 ; Adresse des Statusregisters im RAM-File
9 rp0 equ 5 ; Bank select
10 carry equ 0 ; carry flag - Zeigt an, ob bei der letzten Rechenoperation ein Übertrag auftrat.
11 zero equ 2 ; zero flag - Zeigt an, ob das Ergebnis einer Operation gleich 0 war.
12
13 ; Port A bzw. 0
14 porta equ 5
15 ; Bitstellen von Port A
16 takt equ 0 ; Zähleingang RA0
17 reset equ 1 ; Reseteingang RA1
18 inhibit equ 2 ; Inhibit-Eingang RA2
19 carryOut equ 3 ; Carryausgang RA3
20 maske equ 1 ; 00000001 ; Maske für Zähleingang RA0
21
22 portb equ 6
23 ; TRIS: TRI-State. Mit einem TRIS-Register können Portpins auf Eingang oder Ausgang geschaltet werden.
24 trisa equ 5
25 trisb equ 6
26
27 ; Variablen - Speicherstellen
28 counter equ 10h ; 0c erste freie Adresse, 10h ist im Simulator leicht zu finden
29 aktWert equ 11h
30 alterWert equ 12h
31 flanke equ 13h
32
33 ; \***** labels *****\
34
35     org 0 ; Programm beginnt bei Adresse 0
36
37 cold
38     ; Ports initialisieren
39     bsf status,rp0 ; auf Bank 1 umschalten
40     bcf trisa,carryOut ; Port A carry auf Ausgang setzen
41     clrf trisb ; Port B alles Ausgang
42
43     bcf status,rp0 ; zurück auf Bank 0
44
45     ; ersten Wert lesen
46     movf porta,w ; Port A ins W-file lesen
47     andlw maske ; RA0 Zähleingang maskieren
48     movwf alterWert ; W-file in alterWert Schreiben: erster Vergleichswert
49
50 resetCNT
51     clrf counter ; Counter mit 0 initialisieren
52     bcf porta,carryOut ; carry zurücksetzen
53
54     ; Counterinhalt ausgeben
55     movf counter,w
56     movwf portb
57
58 mainloop
59     btfsc porta,reset ; Resteingang 1?
60     goto resetCNT ; ja → reset
61
62     btfsc porta,inhibit ; Inhibit-Eingang 1?
63     goto mainloop ; ja → Zähler anhalten
64
65     call checkEdge ; Flanke da? Nein → w = 0,
66     xorlw 2 ; setze zero-flag bei steigender Flanke w = 2

```

```

67     btfss status,zero ; w = 2?
68     goto mainloop ; nein
69
70     bcf porta,carryOut ; Ausgabe carry = 0
71     incf counter ; counter++
72     btfsc status,zero ; overflow?
73     bsf porta,carryOut ; ja
74
75     ; Ausgabe counter
76     movf counter,w
77     movwf portb
78
79     goto mainloop
80
81 checkEdge
82     ; aktuellen Wert lesen
83     movf porta,w ; Port A ins w-file lesen
84     andlw maske ; RA0 Zähleingang maskieren
85     movwf aktWert ; w-file in aktWert-File schreiben
86     xorwf alterWert,w ; mit alterWert vergleichen
87     movwf flanke ;
88
89     ; alterWert = aktWert
90     movf aktWert,w
91     movwf alterWert
92
93     ; flanke = 0 wenn aktWert = alterWert
94     movf flanke ; setze zero-flag wenn flanke = 0
95     btfsc status,zero ; flanke 0?
96     retlw 0 ; nein: keine neue Flanke
97
98     movf aktWert ; setze zero-flag wenn flanke = 0
99     btfss status,zero ; aktWert = 0? / zero = 1?
100    retlw 2 ; nein: steigende Flanke
101    retlw 1 ; ja: fallende Flanke

```

```

1 ; BCD-Zähler
2
3     device 16f84
4
5 ; \***** labels *****\
6
7 ; Statusregister
8 status equ 3 ; Adresse des Statusregisters im RAM-File
9 rp0 equ 5 ; Bank select
10 carry equ 0 ; carry flag - Zeigt an, ob bei der letzten Rechenoperation ein Übertrag auftrat.
11 zero equ 2 ; zero flag - Zeigt an, ob das Ergebnis einer Operation gleich 0 war.
12
13 ; Port A bzw. 0
14 porta equ 5
15 ; Bitstellen von Port A
16 takt equ 0 ; Zähleingang RA0
17 reset equ 1 ; Reseteingang RA1
18 inhibit equ 2 ; Inhibitbeingang RA2
19 carryOut equ 3 ; Carryausgang RA3
20 maske equ 1 ; 00000001 ; Maske für Zähleingang RA0
21
22 portb equ 6
23 ; TRIS: TRI-State. Mit einem TRIS-Register können Portpins auf Eingang oder Ausgang geschaltet werden.
24 trisa equ 5
25 trisb equ 6
26
27 bcdOverflowValue equ 10 ; 00001001
28
29 ; Variablen - Speicherstellen

```

```

30 bcdDigit0 equ 10h ; 0c erste freie Adresse, 10h ist im Simulator leicht zu finden
31 bcdDigit1 equ 11h
32 aktWert equ 12h
33 alterWert equ 13h
34 flanke equ 14h
35
36 ; \***** Labels *****\
37
38     org 0 ; Programm beginnt bei Adresse 0
39
40 cold
41     ; Ports initialisieren
42     bsf status,rp0 ; auf Bank 1 umschalten
43     bcf trisa,carryOut ; Port A carry auf Ausgang setzen
44     clrf trisb ; Port B alles Ausgang
45
46     bcf status,rp0 ; zurück auf Bank 0
47
48     ; ersten Wert lesen
49     movf porta,w ; Port A ins W-file lesen
50     andlw maske ; RA0 Zähleringang maskieren
51     movwf alterWert ; W-file in alterWert Schreiben: erster Vergleichswert
52
53 resetCNT
54     clrf bcdDigit0 ; init
55     clrf bcdDigit1 ; init
56     bcf porta,carryOut ; carry zurücksetzen
57
58     clrf portb
59
60 mainloop
61     btfsc porta,reset ; Resteingang 1?
62     goto resetCNT ; ja → reset
63
64     btfsc porta,inhibit ; Inhibit-Eingang 1?
65     goto mainloop ; ja → Zähler anhalten
66
67     call checkEdge ; Flanke da? Nein → w = 0,
68     xorlw 2 ; setze zero-flag bei steigender Flanke w = 2
69     btfss status,zero ; w = 2?
70     goto mainloop ; nein
71
72     bcf porta,carryOut ; Ausgabe carry = 0
73
74     ; BCD erhöhen
75     incf bcdDigit0 ; erste Stelle
76     movf bcdDigit0,w
77     xorlw bcdOverflowValue
78     btfsc status,zero ; overflow bei erster Stelle?
79     call bcd00overflow; ja
80
81     ; BCD Ausgabe
82     swapf bcdDigit1,0 ; ersten vier bits mit den letzten vier bits vertauschen; ,0 → Ergebnis in w
83     ↪ schreiben
84     movwf portb
85     movf bcdDigit0,w
86     addwf portb
87
88     goto mainloop
89
90 bcd00overflow
91     clrf bcdDigit0 ; bcdDigit0 = 0
92     incf bcdDigit1 ; zweite Stelle
93     movf bcdDigit1,w
94     xorlw bcdOverflowValue
95     btfsc status,zero ; overflow bei zweiter Stelle?

```

```

95     call bcd10verflow
96     return
97
98 bcd10verflow
99     clrf bcdDigit1 ; bcdDigit1 = 0
100    bsf porta,carryOut
101    return
102
103 checkEdge
104     ; aktuellen Wert lesen
105     movf porta,w ; Port A ins w-file lesen
106     andlw maske ; RA0 Zähleingang maskieren
107     movwf aktWert ; w-file in aktWert-File schreiben
108     xorwf alterWert,w ; mit alterWert vergleichen
109     movwf flanke ;
110
111     ; alterWert = aktWert
112     movf aktWert,w
113     movwf alterWert
114
115     ; flanke = 0 wenn aktWert = alterWert
116     movf flanke ; setze zero-flag wenn flanke = 0
117     btfsc status,zero ; flanke 0?
118     retlw 0 ; nein: keine neue Flanke
119
120     movf aktWert ; setze zero-flag wenn flanke = 0
121     btfss status,zero ; aktWert = 0? / zero = 1?
122     retlw 2 ; nein: steigende Flanke
123     retlw 1 ; ja: fallende Flanke

```

```

1 ; BCD to seven segment display
2
3     device 16f84
4
5 ; \***** labels *****\
6 pcl equ 2 ; program counter
7 status equ 3 ; status register
8 rp0 equ 5 ; bank select
9 carry equ 0 ; carry flag - Indicates when an arithmetic carry or borrow has been generated out of the
↳ most significant ALU bit position.
10 zero equ 2 ; zero flag - Indicates that the result of an operation was zero.
11
12 ; input: port A input
13 porta equ 5
14 trisa equ 5
15 bcdmask equ Fh ; 00001111
16
17 ; output: port B
18 portb equ 6
19 trisb equ 6
20
21 ; variables
22 bcdin equ 10h
23
24 ; \***** labels *****\
25
26     org 0
27
28 cold
29     ; init
30     bsf status,rp0 ; select bank 1
31     clrf trisb ; set port b to output
32     bcf status,rp0 ; select bank 0
33
34 mainloop

```

```
35     movf porta,w
36     andlw bcdmask
37     movwf bcdin
38     call bcdToSsd
39     movwf portb
40     goto mainloop
41
42 bcdToSsd ; Sevent Segment Display
43     movf bcdin,w
44
45     addwf pcl ; unsave: undefined behavior if bcdin > 9
46     retlw 3Fh ; 0
47     retlw 6h ; 1
48     retlw 5Bh ; 2
49     retlw 4Fh ; 3
50     retlw 66h ; 4
51     retlw 6Dh ; 5
52     retlw 7Dh ; 6
53     retlw 7h ; 7
54     retlw 7Fh ; 8
55     retlw 5Fh ; 9
```