### Schnipsel

```
; /*************** start ***************/
    device 16f84

pcl equ 2 ; program counter
fsr equ 4 ; Indirect data memory address pointer 0.
indf equ 0; Uses contents of FSR to address data memory (not a physical register).

status equ 3
rp0 equ 5 ; bank select
carry equ 0 ; carry flag - Indicates when an arithmetic carry or borrow has been generated out of the most significant ALU bit position.
zero equ 2 ; zero flag - Indicates that the result of an operation was zero.

porta equ 5
ra0 equ 0 ; RA0
portb equ 6

; vars
var equ 10h

    org 0

cold
    bsf status,rp0 ; select bank 1
    ; port initalization
    bcf status,rp0 ; select bank 0
mainloop
; /*************** start ***************/


; /*************** loop ***************/
movlw 4
movwf foo
loop ; foo - 1 iterations
    ; body
    decfsz foo ; decrement f, skip if 0
    goto loop ; not null
; /*************** loop ***************/


; /*************** equals ***************/
movf var1,w
xorwf var2,w
btfsc status,zero
goto equal
goto not_equal
; /*************** equals ***************/


; /*************** number comparison ***************/
movf foo,w
subwf bar,w ; difference
btfsc status,zero ; equal?
goto equal ; yes
; no
btfsc status,carry; (2 complement)
goto fooSmaller ; C = 1 ⟶ foo < bar
goto fooGreater ; C = 0 ⟶ foo > bar
; /*************** number comparison ***************/





; /*************** indirect ***************/
movlw 12h ; adress
movwf fsr
movf indf ; get value in 12h
incf indf
; /*************** indirect ***************/


; /*************** software wait ***************/
; 1 ms = 1000 µs; 1 basic command = 1 µs; commands with jump = 2 µs
; initalization
    movlw 101 ; 1 µs; number of deplay iterations
    movwf deplayCounter ; 1 µs
wait ; 1 Iteration 3µs - 100 iterations
    decfsz deplayCounter ; 1 µs
    goto wait ; 2 µs
    ; time = 2 µs + 100 * 3 µs = 302 µs
; /*************** software wait ***************/


; /*************** check edge ***************/
checkEdge
    ; read current value
    movf porta,w
    andlw mask ; mask clock signal on RA0
    movwf currentValue
    xorwf oldValue,w ; compare with oldValue
    movwf edge

    ; oldValue := currentValue
    movf currentValue,w
    movwf oldValue

    ; edge = 0 if currentValue = oldValue
    movf edge ; set zero-flag if edge = 0
    btfsc status,zero ; edge 0?
    retlw 0 ; no ⟶ no new edge

    movf currentValue ; set zero-flag if currentValue = 0
    btfss status,zero ; currentValue = 0? or zero = 1?
    retlw 2 ; no ⟶ rising edge
    retlw 1 ; yes ⟶ falling edge
; /*************** check edge ***************/
```

### BCD-Zähler

```
; BCD counter © Valerio Cocco

    device 16f84

; \***************** labels **********************\

; status regsiter
status equ 3 ; adress of the status register
rp0 equ 5 ; bank select
carry equ 0 ; carry flag - Indicates when an arithmetic carry or borrow has been generated out of the most significant ALU bit position.
zero equ 2 ; zero flag - Indicates that the result of an operation was zero.

; port a
porta equ 5
clock equ 0 ; clock signal in RA0
reset equ 1 ; reset in RA1
inhibit equ 2 ; inhibit in RA2
carryOut equ 3 ; carry out RA3
mask equ 1 ; 00000001 ; mask for the clock signal on RA0

portb equ 6
; TRIS (TRIState regsiter) Used to define the direction (in/out) of port or pin.
trisa equ 5 ; for port a
trisb equ 6 ; for port b

bcdOverfVal0 equ 10 ; 0000 1010
bcdOverfVal1 equ 0A0h ; 1010 0000
bcdMask equ 0Fh ; 0000 1111, mask for the first bcd digit

; variables
counter equ 10h ; 0c first aviable adress
currentValue equ 12h
oldValue equ 13h
edge equ 14h

; \***************** labels **********************\

    org  0 ; program start at adress 0

cold
    ; initialize ports
    bsf status,rp0 ; select bank 1
    bcf trisa,carryOut ; set carry on port a to out
    clrf trisb ; set port all to out

    bcf status,rp0 ; select bank 0

    ; read first value
    movf porta,w ; read port a in w
    andlw mask ; mask clock signal
    movwf oldValue ; write w regsiter to oldValue: first comparison value

resetCNT
    clrf counter ; init
    bcf porta,carryOut ; reset carry

    clrf portb
    bcf porta,carryOut ; output carry 0

mainloop
    ; output BCD
    movf counter,w
    movwf portb

    btfsc porta,reset ; reset in = 1?
    goto resetCNT ; yes ⟶ reset

    btfsc porta,inhibit ; inhibit = 1?
    goto mainloop ; yes ⟶ pause

    call checkEdge ; edge? no ⟶ w = 0,
    xorlw 2 ; w = 2 (rising redge)? set zero flag if w = 2
    btfss status,zero ; w = 2?
    goto mainloop ; no

    bcf porta,carryOut ; output carry 0

    ; increment BCD
    incf counter ; first digit
    movf counter,w
    andlw bcdMask ; mask first digit
    xorlw bcdOverfVal0 ; overlfow on first digit?
    btfss status,zero
    goto mainloop ; no
    ; yes
    movf counter,w
    xorlw bcdOverfVal0 ; set first nibble to 0
    addlw 10h ; increment second nibble
    movwf counter

    xorlw bcdOverfVal1 ; overflow on second digit?
    btfss status,zero
    goto mainloop; no
    ; yes
    clrf counter
    bsf porta,carryOut

    goto mainloop

checkEdge
    ; read current value
    movf porta,w
    andlw mask ; mask clock signal on RA0
    movwf currentValue
    xorwf oldValue,w ; compare with oldValue
    movwf edge

    ; oldValue := currentValue
    movf currentValue,w
    movwf oldValue

    ; edge = 0 if currentValue = oldValue
    movf edge ; set zero-flag if edge = 0
    btfsc status,zero ; edge 0?
    retlw 0 ; no ⟶ no new edge

    movf currentValue ; set zero-flag if currentValue = 0
    btfss status,zero ; currentValue = 0? or zero = 1?
    retlw 2 ; no ⟶ rising edge
    retlw 1 ; yes ⟶ falling edge
```

### BCD Siebensegment

```
; BCD to seven segment display © Valerio Cocco

    device 16f84

; \***************** labels *********************\
pcl equ 2 ; program counter

status equ 3 ; status register
rp0 equ 5 ; bank select
carry equ 0 ; carry flag - Indicates when an arithmetic carry or borrow has been generated out of the most significant ALU bit position.
zero equ 2 ; zero flag - Indicates that the result of an operation was zero.

; input: port A input
porta equ 5
trisa equ 5
ra6 equ 6
bcdmask equ 0Fh ; 00001111

; output: port B
portb equ 6
trisb equ 6

; variables
bcdin equ 10h
; \***************** labels *********************\

    org 0

cold
    ; init
    bsf status,rp0 ; select bank 1
    clrf trisb ; set port b to output, RB0 is LSB
    bcf trisa,ra6 ; set RA6 to output: digit 0 select
    bcf status,rp0 ; select bank 0

    bsf porta,ra6

mainloop
    movf porta,w
    andlw bcdmask
    movwf bcdin
    call bcdToSsd
    movwf portb
    bcf porta,ra6 ; output portb to digit 0
    bsf porta,ra6 ; lock output from protb to digit
    goto mainloop

bcdToSsd ; Sevent Segtment Display
    movf bcdin,w

    addwf pcl ; unsave: undefined behavior if bcdin > 9
    retlw 3Fh ; 0
    retlw 06h ; 1
    retlw 5Bh ; 2
    retlw 4Fh ; 3
    retlw 66h ; 4
    retlw 6Dh ; 5
    retlw 7Dh ; 6
    retlw 07h ; 7
    retlw 7Fh ; 8
    retlw 6Fh ; 9
    ; retlw 0 ; B
    ; ...
```

### Frequenzteiler

```
; frequency divider © Valerio Cocco

    device 16f84

; \***************** labels **********************\
status equ 3 ; status register
rp0 equ 5 ; bank select
carry equ 0 ; carry flag - Indicates when an arithmetic carry or borrow has been generated out of the most significant ALU bit position.
zero equ 2 ; zero flag - Indicates that the result of an operation was zero.

porta equ 5
output equ 1 ; out an ra1
trisa equ 5

currentValue equ 10h
oldValue equ 11h
edge equ 12h

mask equ 00000001b
; \***************** labels **********************\

cold
    bsf status,rp0 ; select bank 1
    bcf trisa,output
    bcf status,rp0 ; select bank 0

    ; read initial value
    movf porta,w
    andlw mask
    movwf oldValue

mainloop
    call checkEdge
    xorlw 2 ; rising edge?
    btfss status,zero
    goto mainloop ; no
    ; yes
    movlw 10b
    xorwf porta
    goto mainloop

checkEdge
    movf porta,w
    andlw mask
    movwf currentValue
    xorwf oldValue,w
    movwf edge

    ; oldValue := currentValue
    movf currentValue,w
    movwf oldValue

    ; edge = 0 if currentValue = oldValue
    movf edge ; set zero-flag if edge = 0
    btfsc status,zero ; edge 0?
    retlw 0 ; no ⟶ no new edge

    movf currentValue ; set zero-flag if currentValue = 0
    btfss status,zero ; currentValue = 0? / zero = 1?
    retlw 2 ; no ⟶ rising edge
    retlw 1 ; ja ⟶ falling edge
```

### KM Zähler

```
; km counter

    device 16f84

; \***************** labels **********************\
pcl equ 2 ; program counter

status equ 3 ; status register
rp0 equ 5 ; bank select
carry equ 0 ; carry flag - Indicates when an arithmetic carry or borrow has been generated out of the most significant ALU bit position.
zero equ 2 ; zero flag - Indicates that the result of an operation was zero.

fsr equ 4 ; Indirect data memory address pointer 0.

indf equ 0; Uses contents of FSR to address data memory (not a physical register).

; input: port A input
porta equ 5
trisa equ 5
ra0 equ 0
ra1 equ 1
ra2 equ 2
ra3 equ 3

; literals
sigMask equ 00001000b ;

; output: port B
portb equ 6
trisb equ 6


; variables
edge equ 8h
bcdin equ 9h
signalCounter equ 10h
currentVal equ 11h
oldVal equ 12h
mCounter equ 13h
kmCounter equ 16h

; \***************** labels **********************\

    org 0

cold
    ; init ports
    bsf status,rp0 ; select bank 1
    clrf trisb ; set port b to output, RB0 is LSB
    bcf trisa,ra0 ; set RA0 to output: digit 0 select
    bcf trisa,ra1 ; digit 1 select
    bcf trisa,ra2 ; digit 2 select
    bsf trisa,ra3 ; sig in
    bcf status,rp0 ; select bank 0

    clrf portb
    ; lock every digit select
    bsf porta,ra0
    bsf porta,ra1
    bsf porta,ra2

    ; init variables
    clrf signalCounter
    clrf mCounter ; 1x m at 13h
    clrf 14h ; 1xx m
    clrf 15h ; km
    clrf kmCounter ; 1x km at 16h

    ; read first value
    movf porta,w ; read port a in w
    andlw sigMask
    movwf oldVal

outputSsd
    ; output digit 0
    movf 14h,w
    movwf bcdin
    call bcdToSsd
    movwf portb
    bcf porta,ra0
    bsf porta,ra0
    ; digit 1
    movf 15h,w
    movwf bcdin
    call bcdToSsd
    movwf portb
    bcf porta,ra1
    bsf porta,ra1
    ; digit 2
    movf kmCounter,w
    movwf bcdin
    call bcdToSsd
    movwf portb
    bcf porta,ra2
    bsf porta,ra2

mainloop
    ; check signal
    call checkEdge ; edge? no ⟶ w = 0,
    xorlw 2 ; w = 2 (rising redge)? set zero flag if w = 2
    btfss status,zero ; w = 2?
    goto mainloop ; no
    incf signalCounter
    movf signalCounter,w
    xorlw 5 ; counted 5 signals? 5 signals ⟶ 10 m
    btfss status,zero
    goto mainloop

    clrf signalCounter ; reset signal counter
    ; increment meter counter
    movlw mCounter,w ; meter counter
    movwf fsr
bcdinc
    incf indf
    movf indf,w
    xorlw 10 ; bcd overflow?
    btfss status,zero
    goto outputSsd ; no
    ; yes
    clrf indf ; clear overflown bcd digit
    ; increment next digit
    movf fsr,w
    xorlw kmCounter ; out of bounds?
    btfsc status,zero
    goto outputSsd ; yes
    ; no
    incf fsr
    goto bcdinc

```

```
checkEdge
    ; read current value
    movf porta,w
    andlw sigMask ; mask clock signal on RA0
    movwf currentVal
    xorwf oldVal,w ; compare with oldVal
    movwf edge

    ; oldVal := currentVal
    movf currentVal,w
    movwf oldVal

    ; edge = 0 if currentVal = oldVal
    movf edge ; set zero-flag if edge = 0
    btfsc status,zero ; edge 0?
    retlw 0 ; no ⟶ no new edge

    movf currentVal ; set zero-flag if currentVal = 0
    btfss status,zero ; currentVal = 0? or zero = 1?
    retlw 2 ; no ⟶ rising edge
    retlw 1 ; yes ⟶ falling edge

bcdToSsd ; Sevent Segtment Display
    movf bcdin,w

    addwf pcl ; unsave: undefined behavior if bcdin > 9
    retlw 3Fh ; 0
    retlw 06h ; 1
    retlw 5Bh ; 2
    retlw 4Fh ; 3
    retlw 66h ; 4
    retlw 6Dh ; 5
    retlw 7Dh ; 6
    retlw 07h ; 7
    retlw 7Fh ; 8
    retlw 6Fh ; 9
    ; retlw 0 ; A
    ; retlw 0 ; B
    ; ...
```

### Impuls Verlängerer

```
    device 16f84

; \***************** labels **********************\
status equ 3
rp0 equ 5

porta equ 5
sigIn equ 0 ; RA0
sigOut equ 1 ; RA1

; vars
msCounter equ 10h

; \***************** labels **********************\

    org 0

cold
    bsf status,rp0 ; bank 1
    bcf prota,sigOut
    bcf status,rp0 ; bank 0

    goto mainloop
```

```

makeOut
    ; check if ms Counter between 2 & 6
    infc msCounter ; 1 μs
    decfsz msCounter
    goto waitOutInit
    goto mainloop ; decfsz = 0

waitOutInit
    bcs prota,sigOut ; 1μs
    movlw 100 ; 1 μs; number of deplay iterations
    movwf deplayCounter ; 1 μs
waitOut ; iteration = 5 μs
    decfsz deplayCounter ; 1 μs (2 μs)
    goto waitOut ; 2 μs
    nop ; 1 μs
    nop ; 1 μs
    goto makeOut ; 2 μs

mainloop
    bcf prota,sigOut ; sigOut = 0
    btfss porta,0 ; high sig? / 2 μs
    goto mainloop ; no
count
    movlw 31 ; 1 μs; number of deplay iterations
    movwf deplayCounter ; 1 μs
wait ; iteration = 3 μs
    decfsz deplayCounter ; 1 μs (2 μs)
    goto wait ; 2 μs

    ; 100 μs

    btfss porta,0 ; still high?
    goto makeOut ; no

    infc msCounter
    goto count
```

### Lauflicht

```
status equ 3
rp0 equ 5
carry equ 0
zero equ 2
pcl equ 2

porta equ 5
portb equ 6
trisa equ 5
trisb equ 6

cold
    bsf status,rp0 ;Auf Bank1 umschalten
    bsf trisa,0 ;RA0 auf Eingang setzen
    clrf trisb ;PortB alles auf Ausgang
    bcf status,rp0 ;Auf Bank 0 umschalten
    bsf portb,0 ;Port RB0 auf 1 Setzen
start
    btfsc porta,0 ;Wenn signal an Port0 erkannt
    goto loop ;in den Loop springen, ansonsten bei
    goto start ;Start weitermachen

loop
    btfsc status,carry ;Wenn Carry auf 1
    rlf portb ;2 mal shiften
    rlf portb ;andernfalls nur
    goto start ;1 mal shiften
```