

14_06 Appello di settembre

Scrivere un sottoprogramma per un microcontrollore Atmel della famiglia AVR XMEGA in grado di convertire il numero di 4 cifre BCD contenuto in X nel suo valore binario, da lasciare in Y.

Nel caso in cui una delle cifre del dato di partenza abbia una codifica non valida, il risultato deve essere posto a 0.

```
/* Il risultato occupa al massimo 2 byte. Infatti nei casi limite vale 9999 (0x270F)
   L'algoritmo di conversione, oltre a verificare che le 4 cifre
   (nell'ordine u, d, c, m) sono minori di 10, deve eseguire l'operazione u+10d+100c+1000m.
   Volendo usare l'istruzione mul in tutti i casi, si può notare che 1000m=10m*100,
   con entrambi i fattori rappresentabili su 1 byte.
*/
bcd2bin16:
    push R16           //salva i registri di appoggio
    push R17
    push R18
    push R19
    push R20
    push R21
...clr YL             //predispone il risultato a 0
    clr YH
    ldi R16,10         //carica le costanti utili per i prodotti
    ldi R17,100
    mov R18,XL         //carica le 2 cifre meno significative
    andi R18,0x0F      //isola u
    cp R18,R16
    brge bcd2bin16_end //codifica u non valida
    mov R19,XL         //carica le 2 cifre meno significative
    swap R19
    andi R19,0x0F      //isola d
    cp R19,R16
    brge bcd2bin16_end //codifica d non valida
    mov R20,XH         //carica le 2 cifre più significative
    andi R20,0x0F      //isola c
    cp R20,R16
    brge bcd2bin16_end //codifica c non valida
    mov R21,XH         //carica le 2 cifre più significative
    swap R21
    andi R21,0x0F      //isola m
    cp R21,R16
    brge bcd2bin16_end //codifica m non valida
    mov YL,R18         //sistema le unità
    mul R19,R16         //10d in R0
    add YL,R0          //non ci può essere carry
    mul R20,R17        //100c in R1:R0
    add YL,R0
    adc YH,R1
```

```
mul R21,R16      //10m in R0
mov R21,R0
mul R21,R17      //1000m in R1:R0
add YL,R0
adc YH,R1
bcd2bin16_end:
pop R21          //ripristina i registri di appoggio
pop R20
pop R19
pop R18
pop R17
pop R16
ret
```