

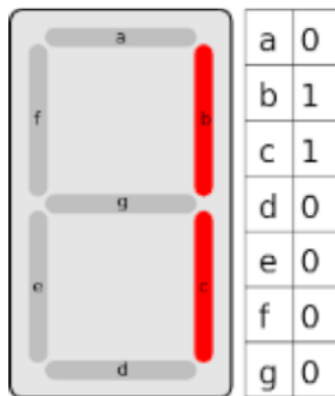
Un processore z64 gestisce il display che indica la MARCIA attualmente ingranata in una motocicletta.

Il processore riceve dalla periferica CAMBIO **una richiesta di interruzione** ogni volta che il guidatore cambia la marcia. Essendo il cambio sequenziale, il processore deve recuperare dal dispositivo CAMBIO l'informazione se il guidatore sta scalando la marcia verso l'alto o verso il basso, leggendo il valore di un opportuno registro di interfaccia.

Il dispositivo MARCIA è equipaggiato con un display a sette segmenti. I valori che possono essere mostrati sono i seguenti:



dove il primo simbolo (n) indica che la moto è a folle. Il dispositivo MARCIA utilizza parole di 7 bit per determinare quali segmenti del led devono essere accesi. Ciascun bit rappresenta lo stato di uno dei segmenti (0 = spento, 1 = acceso), secondo lo schema seguente, dove a è il bit meno significativo:



Realizzare:

- Le interfacce dei dispositivi CAMBIO e MARCIA;
- Tutto il codice necessario al funzionamento del sistema.

SVOLGIMENTO

La periferica CAMBIO (output + IVT) notifica se c'è un'aggiunta di marcia o una diminuzione. La periferica MARCIA è un'interfaccia in Busy Waiting di output con un registro di interfaccia interno che tiene il conto della marcia attualmente ingranata.

.org 0x800
.data

```
.equ IRQ_CAMBIO, 0x0001  
.equ STATUS_CAMBIO, 0x0001  
.equ REG_CAMBIO, 0x0002 #1 scalato verso l'alto, 0 scalato verso il basso  
.equ DISPLAY_MARCIA, 0x0003 #inizialmente a folle (n), #marcia attualmente ingranata  
.equ STATUS_MARCIA, 0x0004  
marcia: .byte 21 #programmo la marcia a folle: 0010101
```

```
.text  
main:  
#inizialmente la marcia attualmente ingranata è "folle"  
outb %al, $STATUS_CAMBIO #avvio la periferica cambio e il processore è disponibile ad essere interrotto.  
call richiedi_marcia  
sti  
hlt
```

richiedi_marcia:

```
.acquisizione:  
outb %al, $STATUS_MARCIA #la periferica marcia viene avviata  
.bw1:  
inb $STATUS_MARCIA, %al  
btb $0, %al  
jnc .bw1  
#il dato è stato prodotto e dentro il registro c'è la marcia attualmente ingranata  
#ora leggiamo dal registro  
inb $DISPLAY_MARCIA, %al #si presume che inizialmente in al ci sia il valore 21  
movb %al, marcia  
#in marcia abbiamo il valore attualmente ingranato della motocicletta  
jmp .acquisizione  
#Nota Bene: Il processore richiede continuamente il valore della marcia
```

```
.driver 0  
push %rcx  
#ora nel driver c'è un aumento o una diminuzione della marcia  
outb %al, $IRQ_CAMBIO #cancello la causa di interruzione  
#ora leggo dal registro di interfaccia della periferica cambio.  
#Se internamente al registro è presente il valore 1 la scalata deve essere effettuata verso l'alto, viceversa il  
valore 0 verso il basso.  
inb $REG_CAMBIO, %al  
movb %al, %cl #in cl è presente il valore 0 (scalata verso il basso) o il valore 1 (scalata verso l'alto).  
call scala  
pop %rcx  
iret
```

```
scala:  
cmpb $0, %cl  
jz .scala_basso  
jnz .scala_alto
```

```
.return:  
outb %al, $STATUS_CAMBIO #la periferica cambio è nuovamente pronta per essere interrotta  
ret
```

```
.scala_basso:  
#se il valore è 0110000 (1.Marcia) diventa 0010101 (folle).  
#se il valore è 1101101 (2.Marcia) diventa 0110000 (1.Marcia)  
#se il valore è 1111001 (3.Marcia) diventa 1101101 (2.Marcia)  
#se il valore è 0110011 (4.Marcia) diventa 1111001 (3.Marcia)  
#se il valore è 1011011 (5.Marcia) diventa 0110011 (4.Marcia)  
cmpb $109, marcia      cmpb $51, marcia  
jz .prima              jz .terza  
cmpb $121, marcia      cmpb $91, marcia  
jz .seconda            jz .quarta  
cmpb $48, marcia  
jz .folle
```

```
.scala_alto:  
#se il valore è folle diventa 0110000 (1. Marcia)  
#se il valore è 0110000 (1.Marcia) diventa 1101101 (2.Marcia)  
#se il valore è 1101101 (2.Marcia) diventa 1111001 (3.Marcia)  
#se il valore è 1111001 (3.Marcia) diventa 0110011 (4.Marcia)  
#se il valore è 0110011 (4.Marcia) diventa 1011011 (5.Marcia)  
cmpb $21, marcia      cmpb $121, marcia  
jz .prima              jz .quarta
```

```

cmpb $48, marcia      cmpb $51, marcia
jz .seconda           jz .quinta
cmpb $109, marcia
jz .terza

```

```

.prima:
    movb $48, marcia
    jmp .return
.seconda:
    movb $109, marcia
    jmp .return
.terza:
    movb $121, marcia
    jmp .return
.quarta:
    movb $51, marcia
    jmp .return
.quinta:
    movb $91, marcia
    jmp .return
.folle:
    movb $21, marcia
    jmp .return

```

Di seguito vengono riportare le interfacce dei dispositivi MARCIA e CAMBIO.

È opportuno ricordare che l'interfaccia CAMBIO richiede che essa gestisca le interruzioni vettorzate poiché il motociclista all'atto del cambio della marcia deve notificare se c'è stata un'aggiunta o meno rispetto a quella attualmente ingranata. Il driver che gestisce l'interruzione deve leggere il valore di un apposito registro di interfaccia interno a CAMBIO e, in base al valore letto, vi sarà un aumento o un decremento della stessa.

La configurazione pensata per l'incremento o il decremento della marcia è la seguente:

- Se il valore letto dal registro di CAMBIO è **0** viene effettuata una scalata verso il **basso**;
- Se il valore letto dal registro di CAMBIO è **1** viene effettuata una scalata verso l'**alto**;

Il codice è stato progettando pensando ad una situazione classica di accensione da parcheggio.

Inizialmente la marcia ingranata non esiste perché il motore è spento;

Appena si accende il motore si parte con la marcia numero uno - dopo che il driver avrà interrotto l'esecuzione corrente del Busy Waiting dell'interfaccia MARCIA (la quale legge continuamente il valore della attuale marcia ingranata) - leggerà il valore uno dal flip flop di CAMBIO andando a verificare la marcia che attualmente è presente, in questo caso "folle" e, alla luce del valore, letto ingranerà la successiva (nell'esempio innescherà la marcia numero 1) tenendo sempre conto del valore della precedente.

Il medesimo procedimento viene effettuato in maniera analoga anche quando il driver legge il valore zero dal flip flop di CAMBIO andando a diminuire la marcia in base a

quella precedentemente innescata.

