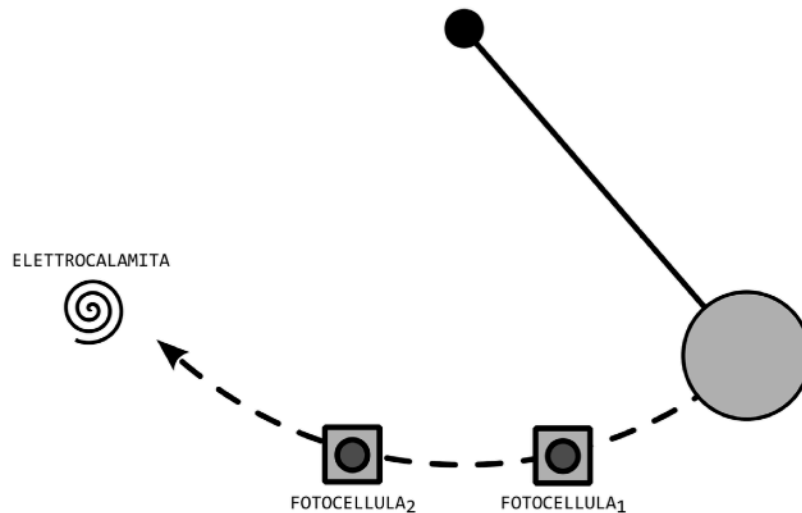


Un processore *z64* controlla il sistema di scappamento elettromeccanico di un orologio a pendolo, come da figura. In tale sistema una coppia di FOTOCELLULE rileva il passaggio del pendolo in due posizioni ben precise, interagendo in maniera asincrona con il processore *z64* .



Il tempo intercorso tra il passaggio davanti a FOTOCELLULA1 e FOTOCELLULA2 viene misurato utilizzando un dispositivo sincrono TIMER . Tale dispositivo, una volta avviato, incomincia a misurare il tempo, alla grana dei millisecondi. Quando successivamente il processore *z64* arresta il dispositivo, il tempo trascorso può essere acquisito mediante un apposito registro di interfaccia.

Una volta determinato il tempo trascorso nel passaggio di fronte alle due FOTOCELLULE , lo *z64* attiva il dispositivo ELETTROCALAMITA per un tempo pari a del tempo di passaggio, al fine di fornire energia addizionale al pendolo—si può utilizzare TIMER anche per misurare questo tempo. Si noti che, essendo presente una sola ELETTROCALAMITA , non è di interesse calcolare il tempo trascorso nel passaggio tra FOTOCELLULA e FOTOCELLULA .

Realizzare:

- Le interfacce delle periferiche FOTOCELLULA , TIMER , ELETTROCALAMITA ;
- Tutto il software necessario al funzionamento del sistema, comprensivo di eventuali driver.

SUGGERIMENTO: l'avvio e l'arresto di TIMER possono essere interpretate come due differenti operazioni.

.org 0x800

.data

```
.equ $IRQ_FOTO_CELLULA1, 0x0001
.equ $IRQ_FOTO_CELLULA2, 0x0002
.equ $STATUS_FOTO_CELLULA_1, 0x0003
.equ $STATUS_FOTO_CELLULA_2, 0x0004
.equ $TEMPO_FOTO_CELLULA_1, 0x0005
.equ $TEMPO_FOTO_CELLULA_2, 0x0006 #(FOTOCELLULA int.input + ivn)
.equ $REG_TIMER #(Timer int.input + busy waiting)
.equ $STATUS_TIMER, 0x0007
.equ $STATUS_ELETTROCALAMITA, 0x0008
.equ $REGISTRO_ELETTROCALAMITA, 0x0009 #dentro c'è il valore di quanto deve essere attivo questo dispositivo
timer: .byte 0 #inizialmente il timer è azzerato
tempo_trascorso: .byte 0 #soma dei tempi
```

.text

main:

```
outb %al, $STATUS_FOTO_CELLULA_1 #avvio foto cellula 1
outb %al, $STATUS_FOTO_CELLULA_2 #avvio foto cellula 2
sti
```

.tempo_acquisizione:

```
outb %al, $STATUS_TIMER #status timer si aggiorna con i millisecondi
```

.bw1:

```
inb $STATUS_TIMER, %al
btb $0, %al #aspetto che il dato in millisecondi venga prodotto
jnc .bw1
inb $REG_TIMER, %al #leggo il valore del timer prodotto dalla PU
movb %al, timer
jmp .tempo_acquisizione
hlt
```

.driver 0

```
push %rcx
outb %al, $IRQ_FOTO_CELLULA1 #abbasso la richiesta di interruzione
movb timer, %cl #in cl ho il tempo trascorso impiegato dalla fotocellula 1
outb %cl, $TEMPO_FOTO_CELLULA_1
addb %cl, tempo_trascorso
outb %al, $STATUS_TIMER #azzero il timer
pop %rcx
iret
```

.driver 1

```
push %rcx
outb %al, $IRQ_FOTO_CELLULA2 #abbasso la richiesta di interruzione
movb timer, %cl #in cl ho il tempo trascorso impiegato dalla fotocellula 2
outb %cl, $TEMPO_FOTO_CELLULA_2
addb %cl, tempo_trascorso #in tempo_trascorso ho il vero valore del tempo trascorso delle due fotocellule
outb %al, $STATUS_TIMER #azzero il timer
call elettrocalamita #attivo la elettrocalamita
pop %rcx
iret
```

elettrocalamita:

```
#elettrocalamita device di input + busy waiting
movb $0, %al
outb %al, $STATUS_TIMER #azzero il timer
```

```
outb tempo_trascorso, $REGISTRO_ELETTROCALAMITA #scrive nel reg.interfaccia il tempo attraverso il quale la
elettrocalamita deve rimanere attiva
```

.loop:

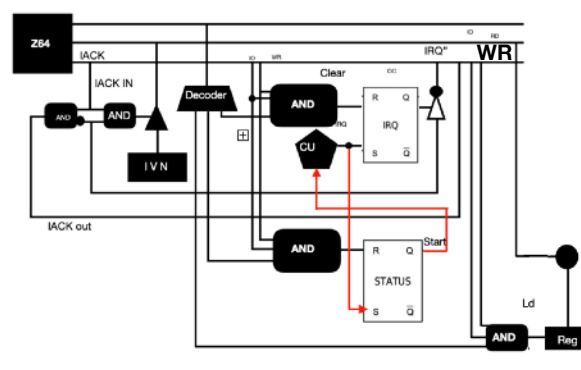
```
outb %al, $STATUS_ELETTROCALAMITA #attivo il dispositivo elettrocalamita (rimane attiva fino a bel)
outb %al, $STATUS_TIMER #richiedo un nuovo dato (timer)
```

.bw2:

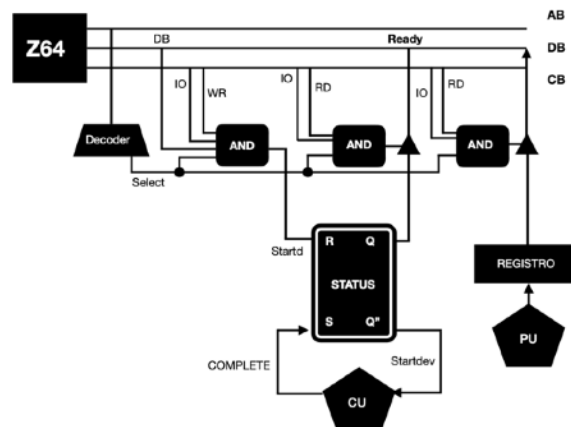
```
inb $STATUS_TIMER, %al
btb $0, %al #aspetto che il dato in millisecondi venga prodotto
jnc .bw2
inb $REG_TIMER, %al #prendo il valore del timer corrente
movb %al, %cl #in cl c'è il timer che attualmente sta scorrendo
inb $REGISTRO_ELETTROCALAMITA, %bpl #in bpl ho il tempo trascorso delle due fotocellule
cmpb bel, %cl #se non siamo arrivati al tempo trascorso continua a scorrere il timer
jnz .loop
movb $0, %al
outb %al, $STATUS_TIMER #azzero il timer
outb %al, $STATUS_FOTO_CELLULA_1 #la fotocellula 1 è nuovamente pronta per essere interrotta
outb %al, $STATUS_FOTO_CELLULA_2 #la fotocellula 2 è nuovamente pronta per essere interrotta
ret
```

Di seguito le interfacce dei dispositivi:

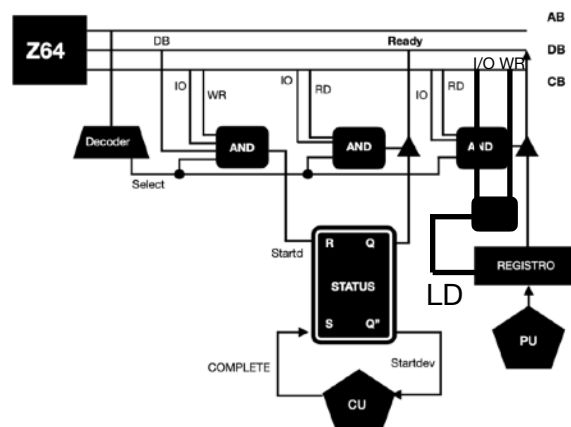
Fotocellula - INTERRUPT OUTPUT perché il processore deve trasferire i dati sul registro interno di ciascuna fotocellula per aggiornare il tempo trascorso.



Timer - interfaccia di input in Busy Waiting per leggere il valore del tempo trascorso processato dalla PU. (Astrazione)



Elettrocalamita - Intf. Input/output in Busy Waiting poiché lo z64 deve trasferire dal registro sul bus dati le informazioni relative al tempo trascorso.



Le unità di processing del timer e della elettrocalamita producono entrambe un valore in millisecondi auto-increment ogni volta che si richiede l'acquisizione di un nuovo dato.