

Monitoraggio stanza.

Una stanza è monitorata da quattro sensori di temperatura, che sono pilotati da un processore z64. Quest'ultimo controlla costantemente che il valor medio della temperatura rilevata nella stanza sia compreso nell'intervallo $[T_{min}, T_{max}]$. Nel caso in cui la temperatura non cada all'interno di questo intervallo, il microprocessore invia un segnale di allarme ad un'apposita periferica (ALLARME). Il segnale d'allarme utilizzato è il valore 1 codificato con 8 bit. Se la temperatura ritorna all'interno dell'intervallo $[T_{min}, T_{max}]$, la CPU trasmette alla periferica il valore 0. I sensori restituiscono la temperatura misurata come un intero a 16 bit, utilizzando i decimi di grado Celsius come unità di misura.

Scrivere il codice assembly per il controllo dei sensori di temperatura e della periferica ALLARME, utilizzando il meccanismo delle interruzioni vettorizzate.

SVOLGIMENTO

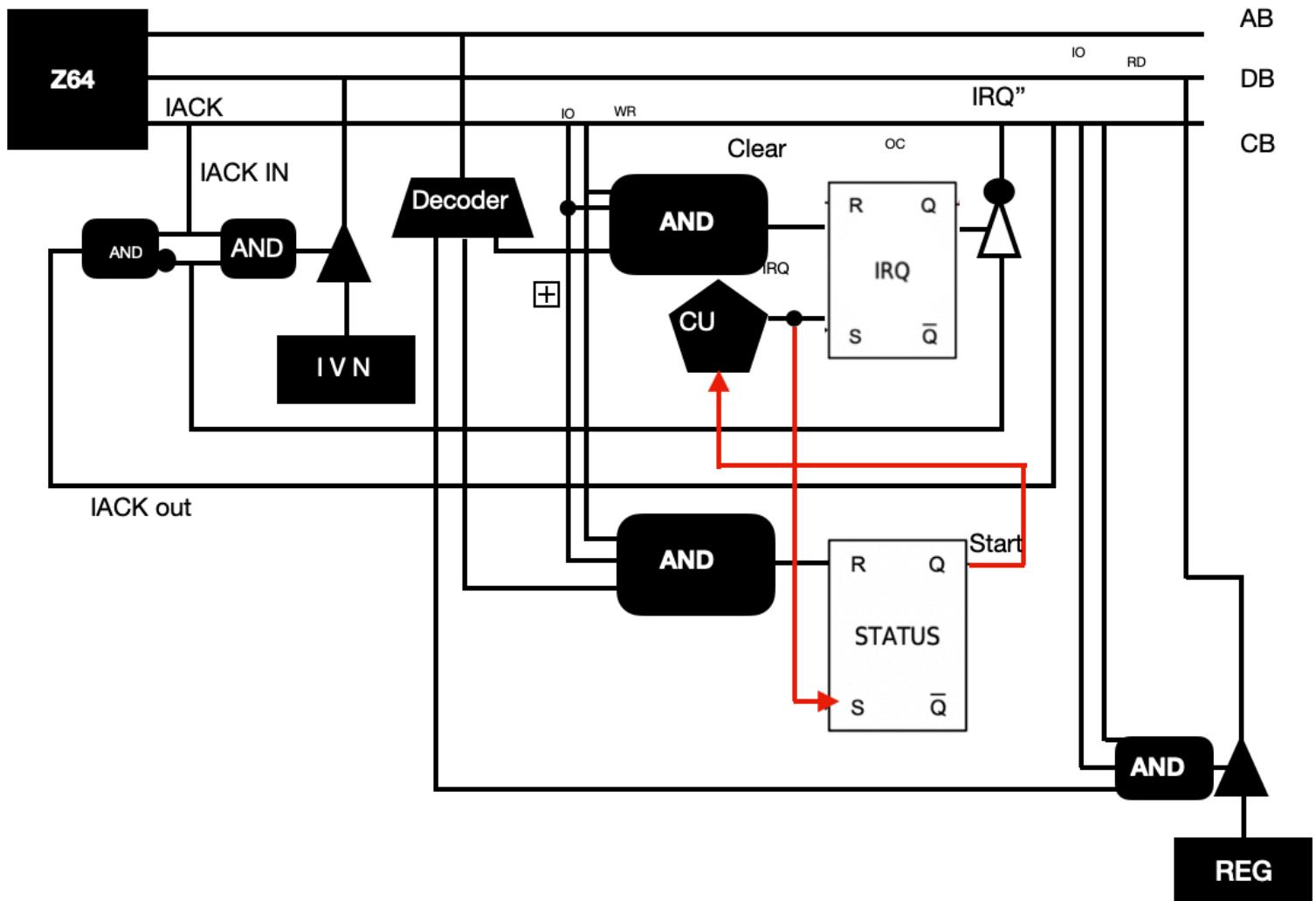
Prima di mostrare le interfacce dei vari dispositivi è opportuno considerare

alcune scelte di progetto.

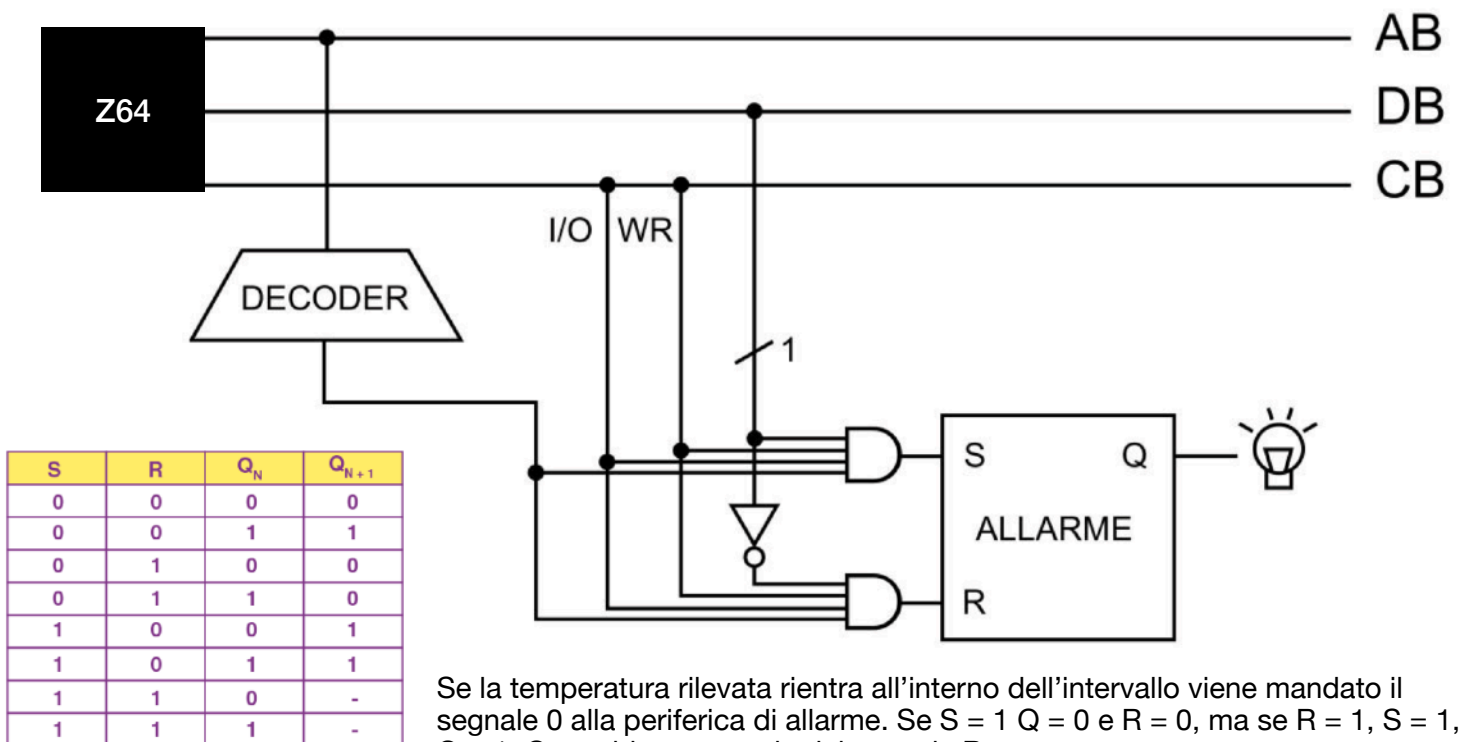
- Le misure di temperatura dei quattro sensori vengono memorizzate all'interno di un vettore di quattro elementi;
- All'avvio del sistema, le quattro misure vengono forzate al centro dell'intervallo $[T_{min}, T_{max}]$;
- Il sensore è una periferica di input che fornisce un solo registro di sola lettura che contiene il valore misurato;
- Se la temperatura è negativa, il sensore restituisce comunque il valore 0;
- ALLARME è una periferica di output, che attiva/disattiva una "sirena" lampeggiante. Un flip flop collegato al bit meno significativo del bus dati accende/spegne l'allarme quando viene impostato/reimpostato.
- ALLARME è una periferica passiva: non ha CU (Control Unit).

SENSORE è una periferica che possiede un registro contenente il valore misurato in gradi celsius. In modo particolare deve essere una periferica di input poiché il processore deve poter leggere il suo contenuto.

SENSORE



ALLARME



Di seguito il codice completo:

```
.org 0x800
.data
    temperature: .word 250, 250, 250, 250
    .equ SENSORE_1_REG, 0x0
    .equ SENSORE_2_REG, 0x1
    .equ SENSORE_3_REG, 0x2
    .equ SENSORE_4_REG, 0x3
    .equ SENSORE_1_STATUS, 0x4
    .equ SENSORE_2_STATUS, 0x5
    .equ SENSORE_3_STATUS, 0x6
    .equ SENSORE_4_STATUS, 0x7
    .equ SENSORE_1_IRQ, 0x8
    .equ SENSORE_2_IRQ, 0x9
    .equ SENSORE_3_IRQ, 0xa
    .equ SENSORE_4_IRQ, 0xb
    .equ tMin, 200
    .equ tMax, 300
    .equ allarme, 0xc

.text
main:

    xorw %r8w, %r8w
    xorw %r9w, %r9w
    outb %al, $SENSORE_1_STATUS #avvio i sensori
    outb %al, $SENSORE_2_STATUS #avvio i sensori
    outb %al, $SENSORE_3_STATUS #avvio i sensori
    outb %al, $SENSORE_3_STATUS #avvio i sensori
    sti

.loop:
    movw %r8w, %r9w #r9w = r8w vecchia media = nuova media
    call media
    movw %cx, %r8w
    cmpw %r8w, %r9w #la media è cambiata?
    jz .loop #se sono uguali e quindi la media non è cambiata allora non faccio nulla

    movb $1, %al #allarme acceso
    cmpw $tMax, %r8w
    jnc .set #sorgente < destinazione = tMax < r8w, bisogna settare ad 1 l'allarme

    cmpw $tMin, %r8w #sorgente > destinazione = tMin > r8w bisogna settare ad 1 l'allarme perché non rientra in [200,300]
    jc .set
    xorb %al, %al #altrimenti allarme spento

.set:
    outb %al, $allarme #imposta l'allarme
    jmp .loop
media:
    push %rbp #callee-save dentro una funzione (Calling Conventions)
    xorq %rbp, %rbp
    xorq %rcx, %rcx

.label:
    movq $temperature, %rdx #rdx contiene l'indirizzo iniziale del vettore
    movq (%rdx, %rbp, 2), %rax
    addq %rax, %rcx
    addq $1, %rbp
    cmpq $4, %rbp
    jnz .label
    shrq $2, %rcx #in rcx ho la media
    pop %rbp
    ret

.driver 1
    #nel driver bisogna inserire il valore letto nel vettore: nella cella corrispondente al driver i-esimo (Semafori). (I 4 driver sono implementati ugualmente)
    push %rdi
    push %rsi
    push %rbx
    movq $SENSORE_1_REG, %rdi #scrive 1 su rdi
    movq $SENSORE_1_STATUS, %rsi
    movq $SENSORE_1_IRQ, %rbx
    call get
    pop %rbx
    pop %rsi
    pop %rdi
    iret
get: #la subroutine recupera da un sensore la temperatura misurata e la memorizza nella cella dell'array corrispondente
    push %rax #serve a contenere la temperatura
    push %rdx #dobbiamo calcolare a run-time l'indirizzo del dispositivo da cui leggere la temperatura perciò usiamo la rappresentazione parametrica
    movw %bx, %dx #prendo il flip flo IRQ e lo metto in dx
    outb %al, %dx #cancello la causa dell'interruzione
    movw %di, %dx #prendo il valore del registro e lo metto in dx
    inw %dx, %ax #leggo il valore dal registro del sensore
    movw %ax, $temperature(, %rdi, 2) #carico il valore nella cella corrispondente
    movw %si, %dx #prendo il valore dello status e lo metto parametrico in dx
    outb %al, %dx #riavvio il dispositivo
    pop %rdx
    pop %rax
    ret
```