19_07_2023 PROGETTO

Un processore z64 gestisce il sistema di controllo di un condizionatore d'aria. Mediante la periferica sincrona TERMOMETRO è in grado di leggere la temperatura corrente della stanza, rappresentata come byte senza segno.

L'utilizzatore del condizionatore ha a disposizione un telecomando ad infrarossi che consente di impostare la temperatura desiderata e la modalità operativa (ventilatore, deumidificatore, condizionatore). **Alla pressione del tasto di conferma, il telecomando trasmette la configurazione desiderata ad una periferica RICEVITORE che avverte in modalità asincrona lo z64 del cambio di impostazione richiesto**.

La periferica RICEVITORE ha a disposizione un registro di interfaccia tramite cui è possibile recuperare la configurazione richiesta. Tale configurazione è una parola di 8 bit, così composta:

- Il bit 7 (più significativo) è un flag booleano che indica che si vuole attivare la modalità ventilatore;
- Il bit 6 è un flag booleano che indica se si vuole attivare la modalità deumidificatore;
- Il bit 5 è un flag booleano che indica se si vuole attivare la modalità condizionatore;
- I 5 bit meno significativi rappresentano un intero senza segno che indica la temperatura richiesta.

Ad esempio, questa configurazione indica l'accensione del deumidificatore per raggiungere una temperatura di 23 GRADI:

7	6	5	4	3	2	1	0
0	1	0	1	0	1	1	1

Alla ricezione della richiesta di interruzione da parte di RICEVITORE , il processore z64 recupera la configurazione richiesta ed attiva/disattiva le periferiche sincrone VENTILATORE , DEUMIDIFICATORE e CONDIZIONATORE , a seconda della richiesta dell'utente. La periferica VENTILATORE ha a disposizione un registro di interfaccia di dimensione byte che consente di specificare la velocità di rotazione, secondo il seguente algoritmo:

- Se la differenza tra la temperatura attuale e quella richiesta è < 4 gradi, il ventilatore viene attivato a velocità 1;
- Se la differenza tra la temperatura attuale e quella richiesta è compresa tra 4 e 8 gradi , viene attivato a velocità 2;
- Altrimenti viene attivato a velocità 3.

Realizzare:

- Le interfacce delle periferiche TERMOMETRO , RICEVITORE , VENTILATORE (NON è necessario realizzare le interfacce delle periferiche DEUMIDIFICATORE e CONDIZIONATORE);
- Tutto il software necessario al funzionamento del sistema, comprensivo di eventuali driver.

S V O L G I M E N T O

```
.org 0x800
.data
        .equ $IRQ_RICEVITORE, 0x0001
        .equ $STATUS_RICEVITORE, 0x0002
        .equ $REGISTRO_RICEVITORE_CONFIGURAZIONE, 0x0003 #ricevitore inter.input + IVN
        #dentro è presento una parola di 8 bit che indica la configurazione richiesta
        .equ $REG_VENTILATORE, 0x0004 #Interfaccia di output (Lo z64 trasferisce i dati nel reg)
#che è uguale a deumidificatore e condizionatore (stesse interfacce)->non messe.
        .equ $STATUS_VENTILATORE, 0x0005 #avvio la periferica ventilatore con questo flip flop
        .equ $STATUS_TERMOMETRO, 0x0006 #interf. Input + busy waiting perch il processore legge la
temperatura
        .equ $REG_TERMOMETRO, 0x0007
        temperatura_attuale: .byte 20
        temperatura_richiesta: .byte 0
        configurazione_attuale: .byte 0

.text
main:
        outb %al, $STATUS_RICEVITORE #Ricevitore può essere interrotto
        sti
    .acquisizione:
        outb %al, $STATUS_TERMOMETRO #avvio la periferica termometro
    .bw1:
        inb $STATUS_TERMOMETRO, %al
        btb $0, %al
        jnc .bw1
        inb $REG_TERMOMETRO, %al
        movb %al, temperatura_attuale #ho la temperatura attuale
        jmp .acquisizione
        hlt

        .driver 0
        push %rcx
        outb %al, $IRQ_RICEVITORE #cancello la causa di interruzione
```

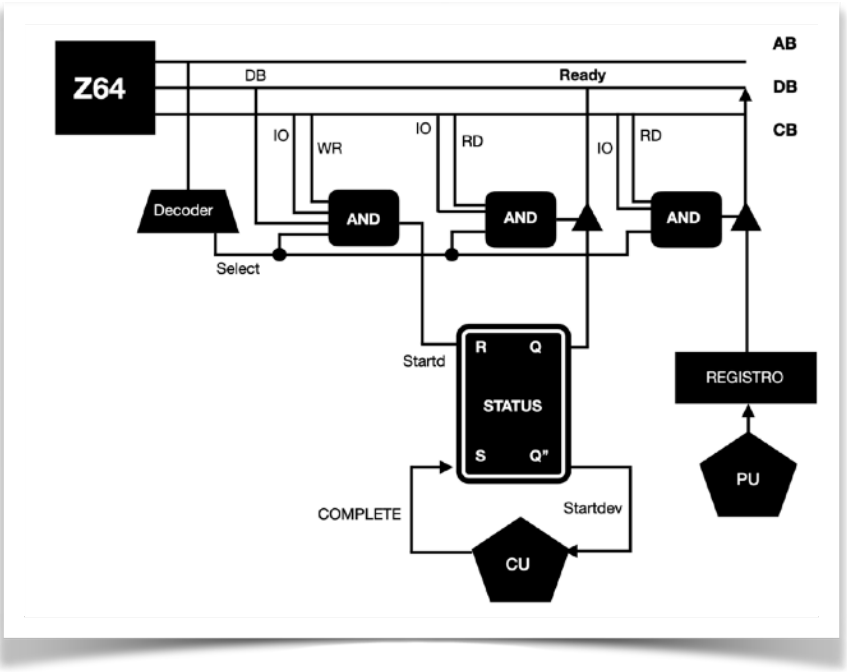
```
        inb $REGISTRO_RICEVITORE_CONFIGURAZIONE, %al
        movb %al, configurazione_attuale
        #bisogna testare i bit che sono in configurazione_attuale
        #testo se si vuole attivare modalità ventilatore effettuando un test con maschera 10000000
        call testing
        pop %rcx
        iret

        testing:
                testb $0x80, configurazione_attuale
                jnz .ventilatore_on #significa che il bit 7 è impostato
                testb $0x40, configurazione_attuale
                jnz .deumidificatore_on #significa che il bit 6 è impostato
                testb $0x20, configurazione_attuale
                jnz .condizionatore_on #significa che il bit 5 è impostato
        .deumidificatore_on:
                outb %al, $STATUS_DEUMIDIFICATORE #non richiesta dal problema - inserita per completezza.
        .condizionatore_on:
                outb %al, $STATUS_DEUMIDIFICATORE #non richiesta dal problema - inserita per completezza.
        .ventilatore_on:
                #ora leggo la temperatura richiesta
                movb configurazione_attuale, %cl
                andb $0x1F, %cl #bit-extraction last five -> degree here = temperatura richiesta
                outb %al, $STATUS_VENTILATORE #attivo il ventilatore ed informo la periferica che c'è un dato da
consumare
        .loop:
                inb $STATUS_VENTILATORE, %al
                btb $1, %al #logica negata per interfacciamento output
                jc .loop
                #ora posso scrivere dentro al ventilatore
                movb temperatura_attuale, %sil #in sul c'è la temperatura attuale
                subb %sil, %cl #in cl c'è la differenza delle due temperature
                cmpb $4, %cl #se la destinazione (cl) è minore della sorgente(4) = sorg > dest
                jc .attiva_1
                #se la destinazione è maggiore della sorgente = differenza_temperature > 4
                jnc .procedura_setting #qui verifichiamo se è minore di 8

        .attiva_1:
                outb $1, $REG_VENTILATORE #Imposto l'attivazione della ventola ad 1
                jmp .return
        .attiva_2:
                outb $2, $REG_VENTILATORE #Imposto l'attivazione della ventola a   2
                jmp .return
        .attiva_3:
                outb $3, $REG_VENTILATORE #imposto l'attivazione della ventola a   3
                jmp .return
        .procedura_setting: #ho saltato perché è maggiore di 4, ma è minore di 8?
                cmpb $8, %cl #destinazione < sorgente  = sorgente > destinazione
                jc .attiva_2 #se è minore di 8 attiva_2
                jmp .attiva_3 #se è maggiore di 8 [e maggiore di 4 (incluso)]
        .return:
                outb %al, $IRQ_RICEVITORE #ricevitore è disponibile per essere nuovamente interrompibile
                ret
```

Di seguito vengono implementate le interfacce delle periferiche.

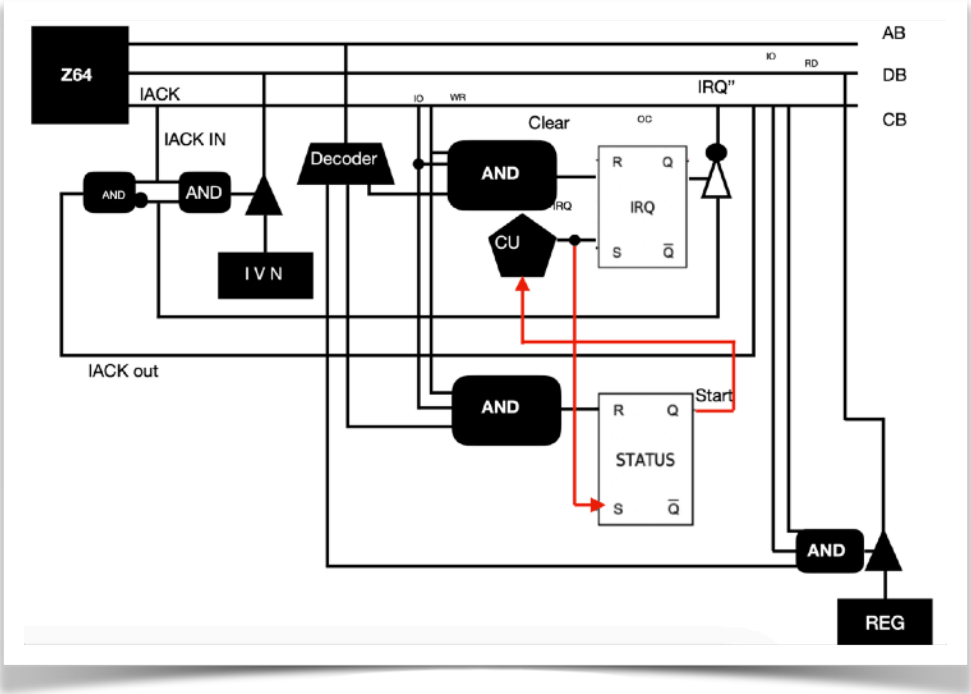
Termometro è un'interfaccia di input in Busy Waiting poiché leggerà opportunamente un nuovo valore e lo trasferirà al processore (utilizzando un'istruzione di IN) in un ciclo indefinito.

Il valore che viene trasferito è la temperatura attuale (o corrente) della stanza.

Interfaccia Termometro - BusyWaiting + input

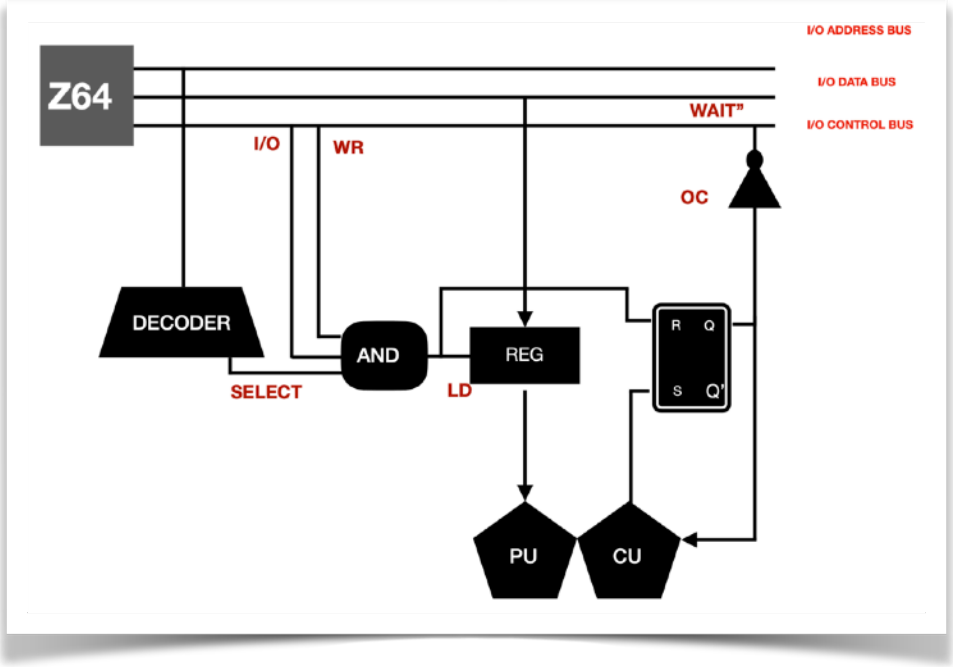


Interfaccia Ricevitore - IVN + input



L’interfaccia ricevitore funziona con le interruzioni vettorizzate. In modo particolare quando arriva un interrupt all’interno del registro di interfaccia sarà presente la configurazione attuale della gestione del condizionatore, in modo particolare quando l’utente preme il tasto del telecomando ad infrarossi per cambiare lo stato della configurazione. Per questo bisogna leggere il valore del registro motivo per il quale essa è un’interfaccia di input.

Interfaccia Ventilatore - output



L'interfaccia del ventilatore è semplicemente un'output generico.

Il processore deve scrivere i dati dentro al suo registro di interfaccia.

I dati che devono essere scritti mediante l'ausilio di una out rappresentano i valori corrispondenti all'attivazione della ventola.

Si presume che poi questi dati vengano letti e processati in modo tale da attivare la ventola.

Nell'esempio il codice mostrato si limita a scrivere il valore di attivazione all'interno del registro senza preoccuparsi poi di "come" effettivamente questa venga attivata (si presume tramite l'ausilio di collegamenti meccanici di cui non si è trattato nella progettazione).

Lo stesso discorso vale per il termometro: si presume che la periferica abbia un sensore di temperatura che legga continuamente un valore e lo scriva nel suo registro di interfaccia;

Il codice proposto non si occupa della parte meccanica che si occupa di come reperire il dato e di come scriverlo nel registro, anzi, si occupa della lettura di **quel dato**, motivo per il quale si parla di interfacciamento di input.