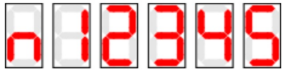
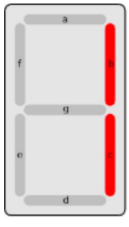
- Un processore z64 gestisce il display che indica la `MARCIA` attualmente ingranata in una motocicletta.

  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:

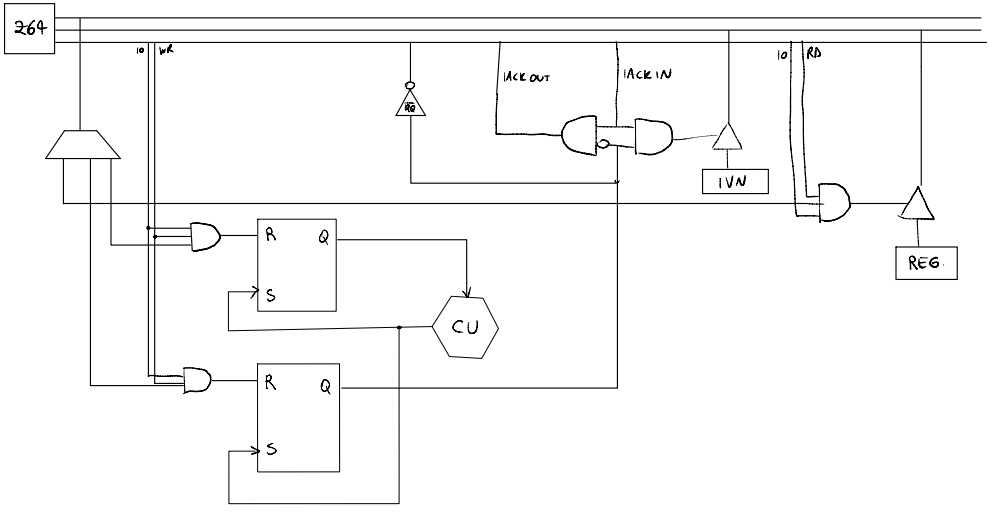
  

a	0
b	1
c	1
d	0
e	0
f	0
g	0

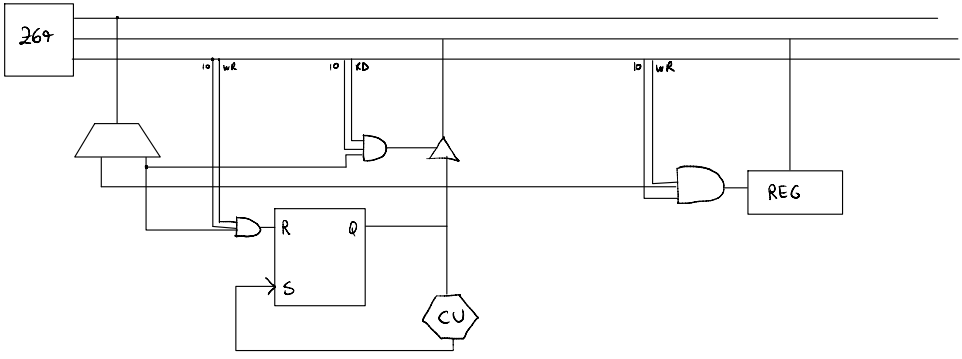
  Realizzare:

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

INTERFACCIA CAMBIO



INTERFACCIA MARCIA



Decido di usare un vettore per mantenere le codifiche delle marce:

Π = 1010100 = 84
1 = 0000110 = 6
2 = 1011011 = 91
3 = 1111001 = 79
4 = 0110011 = 102
5 = 1011011 = 109

CODICE

```
.data
  .equ CAMBIO_status, 0x0
  .equ CAMBIO_irq, 0x1
  .equ CAMBIO_reg, 0x2
  .equ MARCIA_status, 0x3
  .equ MARCIA_reg, 0x4
 array: .byte 84,6,109, 121,51,91
 current: .byte 0
.text
  main:
    sti
   .stampa:
    outb %al, $MARCIA_STATUS
    movb current, %r8b
    movb array(, r8b, 1), %al
    outb %al, $MARCIA_reg
    .bw1:
      inb $MARCIA_status, %al
      btb $0, %al
      jnc .bw1

    outb %al, $CAMBIO_status
    hlt
    jmp .stampa
```

```
.driver 1
 push %rax
 inb $CAMBIO_reg, %al
 outb %al, $CAMBIO_irq
 Sti
 Cmpb $1, %al
 Jz .aumenta
 cmpb $0, current
 Jz .nulla
 Subb $1, current
 Jmp .nulla
.aumenta:
 cmpb $5, current
 Jz .nulla
 addb $1, current
.fine
.nulla:
 pop %rax
 iret
```