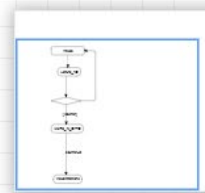
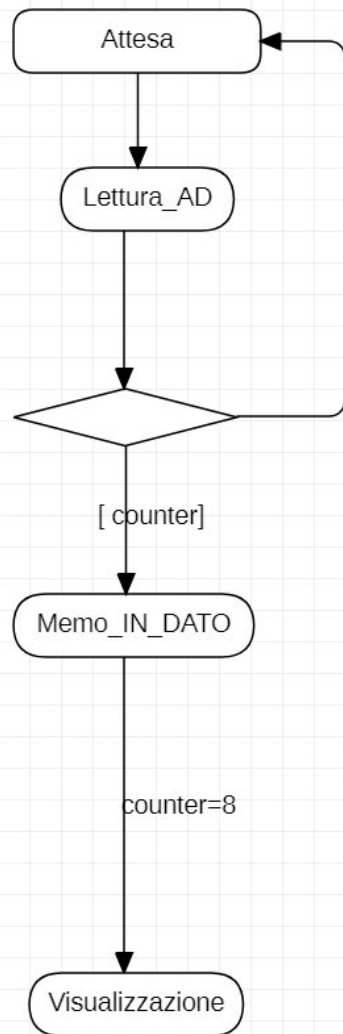


## Esercizio 005



**Il programma controlla** ogni 100ms se il dato è pronto, lo memorizza e incrementa il contatore altrimenti torna in attesa del dato. All'ottavo numero calcola la media e lo visualizza allo schermo.

#caricamento dati da indirizzo 0x10010000

.data

AD: .half 0x1000

IN\_DATO: .half 0x0008

D\_0: .byte 0x0000

D\_1: .byte 0x0000

D\_2: .byte 0x0000

D\_3: .byte 0x0000

#caricamento codice dal indirizzo 0x400000

#inizializzazione dei registri

.text

la \$t0,AD

li \$t1,0x1000

addi \$t2, \$zero,0 # contatore

addi \$s3, \$zero,0 # sum

addi \$t6, \$zero,0 # unita

addi \$t7, \$zero,0 # decine

addi \$t8, \$zero,0 # centinaia

addi \$t9, \$zero,0 # migliaia

#ciclo di lettura dati rimane nel ciclo finché il bit di interesse non è 1

Attesa: lh \$s0,0(\$t0)

and \$s1,\$s0,\$t1

bne \$s1,\$t1,attesa

#conteggio e somma di 8 numeri da leggere

lettura : la \$t3 , IN\_DATO

lh \$s2,0( \$t3 )

add \$s3, \$s3, \$s2

addi \$t2,\$t2,1

slti \$t4,\$t2,8

beq \$t4,\$zero,media

j attesa

#calcolo della media

media:       div \$s3,\$t2

mflo \$s4

li \$t5, 48               # ASCII '0'

add \$t6, \$s4, \$zero   # Copia la media in \$t6

# Estrai cifra delle unità

andi \$t7, \$t6, 0xFF   # Estrai l'ultima cifra

add \$t7, \$t7, \$t5       # Converte la cifra in ASCII

sb \$t7, D\_0            # Scrivi la cifra delle unità al display

# Cifra delle decine

srl \$t6, \$t6, 8        # Shift a destra per ottenere la cifra delle decine

andi \$s5, \$t6, 0xFF   # Estrae la cifra

add \$s5, \$s5, \$t5       # Converte la cifra in ASCII

sb \$s5, D\_1            # Salva la cifra delle decine

# Cifra delle centinaia

srl \$t6, \$t6, 8

andi \$s5, \$t6, 0xFF

add \$s5, \$s5, \$t5

sb \$s5, D\_2

# Cifra delle migliaia

srl \$t6, \$t6, 8

andi \$s5, \$s5, 0xFF

add \$s5, \$s5, \$t5

sb \$s5, D\_3

j stop

stop: j stop