



$\mu_1: \text{DLCHAR} \rightarrow \text{BKR}, \text{INCR}(\text{BKR}) \rightarrow \text{BKR}$   
 $\mu_2: \text{BKR} \rightarrow \text{A}$   
 $\mu_3: \text{SHR}(\text{A}) \rightarrow \text{A}$   
 $\mu_4: A+B \rightarrow B, \text{HCHAR}$   
 $\mu_5: \text{SHL}(\text{A}) \rightarrow \text{A}$   
 $\mu_6: B \rightarrow AC$

## TABELLA ROM

I	Segment P			Segment Q		
	CH(MHE)	NBBs	WkW	WkW	WkW	WkW
G01%	0	0	0.00	0.00	0.00	p1
G02%	1	0.00	0.00	0.00	0.00	p1
G03%	1	0.00	0.00	0.00	0.00	p1
G04%	1	0.00	0.00	0.00	0.00	p1
G05%	-	-	0.00	0.00	0.00	p2
G06%	1	-	0.00	0.00	0.00	p3
G07%	1	0.00	0.00	0.00	0.00	p4
G08%	1	0.00	0.00	0.00	0.00	p5
G09%	1	0.00	0.00	0.00	0.00	p6
G10%	1	0.00	0.00	0.00	0.00	p7
G11%	0	0.00	0.00	0.00	0.00	q1

Esercizio tratto dalla traccia del 12/04/16

Estendere il set di istruzioni della macchina con l'operazione **EOSUM X** definita come segue. All'indirizzo X è memorizzata la lunghezza di un array a sua volta memorizzata a partire dall'indirizzo X+1. L'istruzione calcola la differenza tra la somma degli elementi pari presenti nella prima metà del vettore e somma degli elementi dispari presenti nella seconda metà del vettore. Il valore di tale differenza è memorizzato nell'accumulatore.

IR: COP 1052

## Memory

1052	$n$	7
1053	$V[0]$	12
1054	$V[1]$	5
1055	$V[2]$	4
1056	$V[3]$	10
1057	$V[4]$	7
1058	$V[5]$	3
1058	$V[6]$	6

$$\text{AC} \leftarrow \sum_{i=0, V[i] \in \mathbb{P}}^{\lfloor |V|/2 \rfloor} V[i] - \sum_{i=\lfloor |V|/2+1 \rfloor, V[i] \in \mathbb{P}}^{|V|-1} V[i]$$

STATE INSURANCE

- $\mu_1 : IR_2 \rightarrow MAR, 0 \rightarrow A;$
- $\mu_2 : M[MAR] \rightarrow MBR, INCR(MAR) \rightarrow MAR$
- $\mu_3 : MBR \rightarrow T_1;$
- $\mu_4 : SHR(T_1) \rightarrow T_1, T_1 \rightarrow T_2;$

Dopo l'analisi della prima metà:

MAR: **1986**  
 T1: **9**  
 A: **18**  
 T2: **9**  
 MBR: **9**

Prima metà

```

 $\mu_2 : M[MR] \rightarrow MBR$ 
 $INCR(MAR) \rightarrow MAR$ 
 $if\; MBR_i == 11\; then$ 
     $| \mu_3 : MBR \rightarrow B$ 
     $| \mu_4 : A + B \rightarrow A, DECR(T_1) \rightarrow T_1$ 
     $| DECR(T_2) \rightarrow T_2, goto cl;$ 
 $else$ 
     $| \mu_5 : DECR(T) \rightarrow T_1,$ 
     $| DECR(T_2) \rightarrow T_2, goto cl;$ 
 $end$ 

```

TAG-1 (H10.1)

```

Pseudocode EUSUM(X):
len ← ⌊|X|/2⌋; a ← 0; X ← +;
len2 ← len; len ← len/2;
while len > 0 do
    elem ← M[X];
    if elem pari then
        | a ← a + elem;
    end
    len ← len - len2 - 1; X ← +;
end
while len2 > 0 do
    elem ← M[X];
    if elem dispari then
        | a ← a - elem;
    end
    len2 ← len2 - 1; X ← +;
end

```

### Implementazione del ciclo

```

c1: if  $OR(T_1) == 1$  then
|   goto c1
else
|   ...
end
c2: if  $OR(T_2) == 1$  then
|   goto c2
else
|   A  $\rightarrow$  AC;
end

```

Seconda metà

```

 $\mu_2 : M' \cdot MAR \rightarrow MHR;$ 
 $INCR(MAR) \rightarrow MAR;$ 
if  $MBR_0 == 1$  then
   $\mu_5 : MHR \rightarrow B;$ 
   $\mu_6 : A - B \rightarrow A;$ 
   $DEC(R(T_2)) \rightarrow T_2, gate\ c2;$ 
else
   $\mu_8 : DEC(R(T_2)) \rightarrow T_2, gate\ c2;$ 
end

```

The watermark consists of two lines of text. The top line, 'BERTONI', is in a large, bold, sans-serif font. Below it, 'GAIOLA INGENIERIA INFORMATICA' is written in a smaller, regular, sans-serif font. Both lines of text are oriented diagonally from bottom-left to top-right. They are overlaid on a light gray grid background.

## Soluzione:

```

 $\mu_1 : IR_x \rightarrow MAR, 0 \rightarrow A;$ 
 $\mu_2 : M[MAR] \rightarrow MBR, INCR(MAR) \rightarrow MAR;$ 
 $\mu_3 : MBR \rightarrow T_1;$ 
 $\mu_4 : SHR(T_1) \rightarrow T_1, T_1 \rightarrow T_1;$ 
c1: if  $OR(T_1) == 1$  then
     $\mu_2 : M[MAR] \rightarrow MBR, INCR(MAR) \rightarrow MAR;$ 
    // Se il valore letto è pari
    if  $MBR_0 == 0$  then
         $\mu_5 : MBR \rightarrow B;$ 
         $\mu_6 : A + B \rightarrow A, DECR(T_1) \rightarrow T_1, DECR(T_2) \rightarrow T_2, goto c1;$ 
    else
         $\mu_7 : DECR(T_1) \rightarrow T_1, DECR(T_2) \rightarrow T_2, goto c1;$ 
    end
else
     $\mu_0 : \phi;$ 
end
c2: if  $OR(T_2) == 1$  then
     $\mu_2 : M[MAR] \rightarrow MBR, INCR(MAR) \rightarrow MAR;$ 
    if  $MBR_0 == 1$  then
         $\mu_5 : MBR \rightarrow B;$ 
         $\mu_8 : A - B \rightarrow A, DECR(T_2) \rightarrow T_2, goto c2;$ 
    else
         $\mu_9 : DECR(T_2) \rightarrow T_2, goto c2;$ 
    end
else
     $\mu_{10} : A \rightarrow AC;$ 
end

```

Esercizio tratto dalla traccia del 07/04/05

Estendere il set di istruzioni della macchina con l'operazione **CNTVn X** che, dati due vettori  $V$  e  $W$  di 32 elementi, posti in RAM a partire rispettivamente dagli indirizzi memorizzati nelle locazioni  $M[X]$  e  $M[X+1]$ , restituisce nell'accumulatore il numero di coppie di elementi  $V[i]$  e  $W[i]$  ( $0 \leq i \leq 31$ ) che occupano la stessa posizione nei rispettivi vettori e tali che  $V[i] < W[i]$

Memory:

102	ind1	226
103	ind2	113
...	...	...
113	$W[0]$	-2
114	$W[1]$	4
115	$W[2]$	-1
...	...	...
144	$W[31]$	-3
...	...	...
226	$V[0]$	6
227	$V[1]$	3
228	$V[2]$	4
...	...	...
257	$V[31]$	-1

### Pseudocode CNTVn(X):

```

 $ind1 \leftarrow M[X];$ 
 $ind2 \leftarrow M[X + 1];$ 
 $cnt \leftarrow 0; sz \leftarrow 32;$ 
while  $sz > 0$  do
     $vi \leftarrow M[ind1];$ 
     $wi \leftarrow M[ind2];$ 
    if  $vi < wi$  then
         $cnt += 1;$ 
     $ind1 += 1; ind2 += 1; sz -= 1;$ 

```

$\mu_1 : IR_x \rightarrow MAR;$   
 $\mu_2 : M[MAR] \rightarrow MBR, INCR(MAR) \rightarrow MAR;$   
 $\mu_3 : MBR \rightarrow IND1, M[MAR] \rightarrow MBR;$   
 $\mu_4 : MBR \rightarrow IND2, 32 \rightarrow T1, 0 \rightarrow AC;$

Modifiche architettoniche:

- Aggiunta e collegamento al bus dei registri **IND1** e **IND2**.
- Supporto all'istruzione  $0 \rightarrow AC$
- Supporto all'istruzione  $32 \rightarrow T1$

A <sub>T1</sub>	K <sub>T1</sub> <sup>0</sup>	K <sub>T1</sub> <sup>1</sup>	f
0	—	—	Mem
1	0	0	Bus
1	0	1	INCR
1	1	0	DECR
1	1	1	Load 32

$\mu$	$A_{IR}$	$Z_{IR}$	$A_{AC}$	$K_{AC}$	$A_{MAR}$	$K_{MAR}$	$A_{MBR}$	$S$	$L$	$E$	$A_{T1}$	$K_{T1}^0$	$K_{T1}^1$	$A_{IND1}$	$A_{IND2}$
$\mu_1$	0	—	0	—	1	0	0	0	0	—	0	—	—	0	0
$\mu_2$	0	—	0	—	1	1	1	0	1	—	0	—	—	0	0
$\mu_3$	0	—	0	—	0	—	1	1	0	—	0	—	—	1	0
$\mu_4$	0	—	1	1	1	0	0	0	0	—	1	1	1	0	1

### Implementazione del ciclo

```

c: if  $OR(T1) == 1$  then
     $\mu_5 : IND1 \rightarrow MAR, INCR(IND1) \rightarrow IND1;$ 
     $\mu_6 : M[MAR] \rightarrow MBR, IND2 \rightarrow MAR, INCR(IND2) \rightarrow IND2;$ 
     $\mu_7 : MBR \rightarrow A, M[MAR] \rightarrow MBR;$ 
     $\mu_8 : MBR \rightarrow B;$ 
     $\mu_9 : A - B \rightarrow A;$ 
    if  $A_{31} == 1$  then
         $\mu_{10} : INCR(AC) \rightarrow AC, DECR(T1) \rightarrow T1, goto c;$ 
    else
         $\mu_{11} : DECR(T1) \rightarrow T1, goto c;$ 
    end
else
     $\mu_0 : \phi;$ 

```

### Posso fare questo?

A → MBR → A

```

if  $A_{31} == 1$  then
     $\mu_{10} : INCR(AC) \rightarrow AC;$ 
     $\mu_{11} : DECR(T1) \rightarrow T1, goto c;$ 

```

GAI INGEGNERIA INFORMATICA  
BERTOLINI

**Soluzione:**

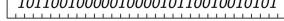
```

 $\mu_1 : IR_x \rightarrow MAR;$ 
 $\mu_2 : M[MAR] \rightarrow MBR, INCR(MAR) \rightarrow MAR;$ 
 $\mu_3 : MBR \rightarrow IND1, M[MAR] \rightarrow MBR;$ 
 $\mu_4 : MBR \rightarrow IND2, 32 \rightarrow T1, 0 \rightarrow AC;$ 
c: if  $OR(T1) == 1$  then
     $\mu_5 : IND1 \rightarrow MAR, INCR(IND1) \rightarrow IND1;$ 
     $\mu_6 : M[MAR] \rightarrow MBR, IND2 \rightarrow MAR, INCR(IND2) \rightarrow IND2;$ 
     $\mu_7 : MBR \rightarrow A, M[MAR] \rightarrow MBR;$ 
     $\mu_8 : MBR \rightarrow B;$ 
     $\mu_9 : A - B \rightarrow A;$ 
    if  $A_{31} == 1$  then
         $\mu_{10} : INCR(AC) \rightarrow AC, DECR(T1) \rightarrow T1, goto c;$ 
    else
         $\mu_{11} : DECR(T1) \rightarrow T1, goto c;$ 
else
     $\mu_0 : \phi;$ 

```

**Esercizio tratto dalla traccia del 07/07/11**

Estendere il set di istruzioni della macchina con l'operazione **Count\_Active X**. A partire dalla locazione di indirizzo  $X$  è presente in RAM un vettore  $V$  di 32 elementi. L'accumulatore contiene un numero a 32 bit. L' $i$ -esimo elemento del vettore  $V$  è attivo se l' $i$ -esimo bit dell'accumulatore vale 1. L'istruzione restituisce nell'accumulatore la somma degli elementi attivi del vettore  $V$ .

**AC:** 

Memory

1052	$V[0]$	9
1053	$V[1]$	-2
1054	$V[2]$	1
1055	$V[3]$	5
1056	$V[4]$	3
1057	$V[5]$	-7
1058	$V[6]$	0
...	...	...
1058	$V[31]$	1

```

 $\mu_1 : IR_x \rightarrow MAR, 32 \rightarrow T1, 0 \rightarrow B;$ 
c: if  $OR(T1) == 1$  then
    if  $AC_{31} == 1$  then
         $\mu_2 : M[MAR] \rightarrow MBR, INCR(MAR) \rightarrow MAR;$ 
         $\mu_3 : MBR \rightarrow A;$ 
         $\mu_4 : A + B \rightarrow B, DECR(T1) \rightarrow T1,$ 
         $SHL(AC) \rightarrow AC, goto c;$ 
    else
         $\mu_5 : INCR(MAR) \rightarrow MAR, DECR(T1) \rightarrow T1,$ 
         $SHL(AC) \rightarrow AC, goto c;$ 
else
     $\mu_6 : B \rightarrow AC$ 

```

**Esercizio tratto dalla traccia del 30/04/14 (riadattamento)**

Estendere il set di istruzioni della macchina con l'operazione **MULT2K X** che verifica se il numero all'indirizzo  $X$  è multiplo di  $2^k$ , dove  $k$  è il valore presente nell'accumulatore. Se la proprietà è verificata l'istruzione restituisce 1 nell'accumulatore, altrimenti restituisce 0.

```

if  $M[X]\%2^k == 0$  then
     $AC \leftarrow 1;$ 
else
     $AC \leftarrow 0;$ 

```

**Soluzione 1**

```

num  $\leftarrow M[X];$ 
while  $AC != 0$  do
    if  $num_0 == 1$  then
         $\quad \text{return } 0;$ 
    num  $\leftarrow SHR(num);$ 
    AC  $\leftarrow AC - 1;$ 
return 1;

```

**Soluzione 2**

Usare una maschera	
0000...0	11...1

$$num \wedge 00000000\dots011\dots1 = \\ \begin{cases} 1 & num_{[0\dots k]} = 0\dots0 \\ 0 & 1 \in num_{[0\dots k]} \end{cases}$$



**Soluzione 1**

```
 $\mu_1 : IR_x \rightarrow MAR, AC \rightarrow T2;$ 
 $\mu_2 : M[MAR] \rightarrow MBR;$ 
 $\mu_3 : MBR \rightarrow T1;$ 
c: if  $OR(T2) == 1$  then
| if  $T1_0 == 1$  then
| |  $\mu_4 : 0 \rightarrow AC;$ 
else
| |  $\mu_5 : SHR(T1) \rightarrow T1,$ 
| |  $DECR(T1) \rightarrow T2, goto c;$ 
| |  $DECR(T2) \rightarrow T2, goto c;$ 
else
| |  $\mu_6 : 1 \rightarrow AC;$ 
```

N.B.  $DECR(T1) \rightarrow T2$  è errato anche da un punto di vista tecnico: il decremento è un'operazione interna al registro, perciò non si può scrivere il risultato del decremento di  $T1$  direttamente in  $T2$ .

**Soluzione 2**

```
 $\mu_1 : IR_x \rightarrow MAR, AC \rightarrow T2, 1 \rightarrow B;$ 
c:if  $OR(T2) == 1$  then
|  $\mu_2 : SHL(B) \rightarrow B,$ 
| |  $DECR(T2) \rightarrow T2, goto c;$ 
else
| |  $\mu_3 : DECR(B) \rightarrow B;$ 
 $\mu_4 : M[MAR] \rightarrow MBR;$ 
 $\mu_5 : MBR \rightarrow A;$ 
 $\mu_6 : A \wedge B \rightarrow T2;$ 
if  $OR(T2) == 0$  then
| |  $\mu_7 : 1 \rightarrow AC;$ 
else
| |  $\mu_8 : 0 \rightarrow AC;$ 
```