

Es 1

Programma che legge due numeri naturali A e B da input e calcola in output il loro minimo
Considerate anche la variante del programma nel caso di registri che possono contenere solo numeri maggiori o uguali a zero.

N istruzione	Istruzione	Commenti
1	READ 1	<i>Leggi il primo input in R1</i>
2	READ 2	<i>Leggi il secondo input in R2</i>
3	LOAD 2	<i>Carica R2 su ACC</i>
4	JGTZ 6	<i>Controlla che il valore sia > 0; nel caso contrario, stop</i>
5	HALT	
6	LOAD 1	<i>Carica R1 su ACC</i>
7	JGTZ 9	<i>Controlla che il valore sia > 0; nel caso contrario, stop</i>
8	HALT	
9	SUB 2	<i>R2-R1 eseguito sull'ACC</i>
10	JGTZ 13	<i>Se R2-R1 > 0, vai a 13</i>
11	WRITE 1	<i>Stampa valore di R1</i>
12	HALT	
13	WRITE 2	<i>Stampa valore di R2</i>
14	HALT	

Es 2

Programma che legge un numero naturale N maggiore o uguale a 0 e quindi calcola il minimo di N valori letti da input.

Nota: Il numero 0 in input serve ad uscire dal programma.

N istruzione	Istruzione	Commenti
1	READ 1	<i>Leggi primo input in R1</i>
2	LOAD 1	<i>Carica su ACC il valore di R1</i>
3	JZERO 11	<i>Se ACC==0, termina</i>
4	JGTZ 6	<i>Se ACC>0, vai a 6</i>
5	HALT	<i>Se ACC<0, termina</i>
6	SUB 2	<i>R2-R1 eseguito sull'ACC</i>
7	JGTZ 10	<i>se R2-R1>0 (ovvero R2>R1), vai a 10.</i>
8	LOAD 1	<i>Se R1>R2, ricarica R1 su ACC...</i>
9	STORE 2	<i>... e salvalo in R2</i>
10	JUMP 1	<i>Ritorna a 1 (ovvero ricomincia il ciclo)</i>
11	WRITE 2	<i>Stampa il valore minimo</i>
12	HALT	<i>STOP</i>

Es 3

Programma che eseguire le stesse operazioni del programma 2 memorizzando però prima gli N valori letti in input in N registri distinti (supponete di avere sempre abbastanza registri per ogni N) e, in una seconda fase effettuando il calcolo del valore minimo usando i dati operazioni sui dati nei registri attraverso l'accumulatore (suggerimento: usate operatore *R che opera sul registro con indirizzo contenuto in R).

Nota:

- R1 = registro contatore
- R2 = registro dove caricare temporaneamente gli input

Ciclo numero 1: salva tutti gli N input su N registri a partire dal 3° compreso

N Istruzione	Istruzione	Commenti
1	ADD=3	Inizializza ACC con "3"
2	STORE 1	Salva valore di ACC su R1
3	READ *1	Leggi input e salva il valore sul RN (N= valore R1) ...
4	LOAD *1	...e sull'ACC
5	JZERO 9	Se input==0, inizia il controllo del numero minore a 9
6	LOAD 1	Carica su ACC il contatore (R1)
7	ADD=1	Aggiungi 1 all'ACC (R1+1)...
8	JUMP 2	...e ricomincia il ciclo di lettura degli input

Inizializzazione di R2 con il valore dell'ultimo registro disponibile

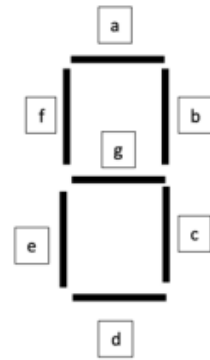
9	LOAD 1	Carica il numero dell'ultimo registro+1 su ACC
10	SUB=1	Rimuovi 1 al ACC per avere il numero dell'ultimo registro
11	STORE 2	Salva temporaneamente il numero del registro a cui puntare (Num) su R2
12	LOAD *2	Carica su ACC il valore del registro Num (ovvero l'ultimo disponibile)
13	STORE 2	Salva il valore su R2

Ciclo numero 2: trova il valore minimo

14	LOAD 1	<i>Carica il numero dell'ultimo registro+1 su ACC</i>
15	SUB=1	<i>Rimuovi 1 al ACC per avere il numero dell'ultimo registro</i>
16	STORE 1	<i>Salva il valore di ACC su R1</i>
17	SUB=3	<i>Rimuovi 3 al ACC...</i>
18	JZERO 25	<i>...per controllare se si è arrivati all'ultimo registro che contiene input (ovvero R3). Nel caso, stampa il val min e termina il programma.</i>
19	LOAD *1	<i>Sovrascrivi ACC con il valore di RN (N= valore R1)</i>
20	SUB 2	<i>ACC-R2</i>
21	JGTZ 14	<i>e $RN-R2 > 0$ ($== RN > R2$) ricomincia il ciclo</i>
22	LOAD *1	<i>Se $RN < R2$ bisogna salvarlo e sovrascrivere R2, quindi carica il valore RN su ACC</i>
23	STORE 2	<i>Salva il valore di ACC su R2</i>
24	JUMP 14	<i>Ricomincia il ciclo</i>
25	WRITE 2	<i>Stampa il valore minimo (salvato su R2)</i>
26	HALT	<i>STOP</i>

Circuiti combinatori

Si vuole costruire un dispositivo elettronico con tre ingressi (i_0, i_1, i_2) e 7 output (a,b,c,d,e,f,g) per controllare un display a led costituito da 7 segmenti (identificati dalle lettere a,b,c,...g) come nella figura a destra. (Nota: consideriamo solo 3 ingressi per semplificare l'esercizio)



Es 1

Costruire la tabella di verità (ovvero la funzione Booleana) per fare in modo che prendendo in ingresso la codifica binaria su 3 bit (con in 0 cifra meno significativa) di una cifra da 0 a 7 vengano messi a 1 i segnali in uscita per visualizzare correttamente la cifra sul display.

Nota: 6 e 9 con 3 segmenti orizzontali.

(N) ₁₀	i1	i2	i3	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
1	0	0	1	0	1	1	0	0	0	0
2	0	1	0	1	1	0	1	1	0	1
3	0	1	1	1	1	1	1	0	0	1
4	1	0	0	0	1	1	0	0	1	1
5	1	0	1	1	0	1	1	0	1	1
6	1	1	0	1	0	1	1	1	1	1
7	1	1	1	1	1	1	0	0	0	0

Es 2

Calcolare la forma normale disgiuntiva (somma di prodotti) e congiuntiva (prodotto di somme) per ogni uscita (ovvero segmento) a,b,c...g.

Guardando la tabella di verità dell'esercizio 1, possiamo prendere i valori positivi (1) dalle colonne a...g per ottenere la formula di ogni uscita espressa come una somma di prodotti; prendendo i valori falsi (0) dalla tabella, otteniamo la formula di ogni uscita espressa come un prodotto di somme:

- a:
 - $a = \overline{i_1} \cdot \overline{i_2} \cdot \overline{i_3} + \overline{i_1} \cdot i_2 \cdot \overline{i_3} + \overline{i_1} \cdot i_2 \cdot i_3 + \overline{i_1} \cdot \overline{i_2} \cdot i_3 + \overline{i_1} \cdot i_2 \cdot \overline{i_3} + \overline{i_1} \cdot \overline{i_2} \cdot i_3$
 - $a = (\overline{i_1} + \overline{i_2} + \overline{i_3}) \cdot (\overline{i_1} + i_2 + \overline{i_3})$

- b:
 - $b = \overline{i1} \cdot \overline{i2} \cdot \overline{i3} + \overline{i1} \cdot \overline{i2} \cdot i3 + \overline{i1} \cdot i2 \cdot \overline{i3} + \overline{i1} \cdot i2 \cdot i3 + i1 \cdot \overline{i2} \cdot \overline{i3} + i1 \cdot \overline{i2} \cdot i3$
 - $b = (\overline{i1} + i2 + \overline{i3}) \cdot (\overline{i1} + \overline{i2} + i3)$
- c:
 - $c = \overline{i1} \cdot \overline{i2} \cdot \overline{i3} + \overline{i1} \cdot \overline{i2} \cdot i3 + \overline{i1} \cdot i2 \cdot \overline{i3} + i1 \cdot \overline{i2} \cdot \overline{i3} + i1 \cdot \overline{i2} \cdot i3 + i1 \cdot i2 \cdot \overline{i3} + i1 \cdot i2 \cdot i3$
 - $c = (i1 + \overline{i2} + i3)$
- d:
 - $d = \overline{i1} \cdot \overline{i2} \cdot \overline{i3} + \overline{i1} \cdot i2 \cdot \overline{i3} + \overline{i1} \cdot i2 \cdot i3 + i1 \cdot \overline{i2} \cdot i3 + i1 \cdot i2 \cdot \overline{i3}$
 - $d = (i1 + i2 + \overline{i3}) \cdot (\overline{i1} + i2 + i3) \cdot (\overline{i1} + \overline{i2} + \overline{i3})$
- e:
 - $e = \overline{i1} \cdot \overline{i2} \cdot \overline{i3} + \overline{i1} \cdot i2 \cdot \overline{i3} + i1 \cdot i2 \cdot \overline{i3}$
 - $e = (i1 + i2 + \overline{i3}) \cdot (i1 + \overline{i2} + \overline{i3}) \cdot (\overline{i1} + i2 + i3) \cdot (\overline{i1} + \overline{i2} + \overline{i3}) \cdot (\overline{i1} + \overline{i2} + \overline{i3})$
- f:
 - $f = \overline{i1} \cdot \overline{i2} \cdot \overline{i3} + i1 \cdot \overline{i2} \cdot \overline{i3} + i1 \cdot \overline{i2} \cdot i3 + i1 \cdot i2 \cdot \overline{i3}$
 - $f = (i1 + i2 + \overline{i3}) \cdot (i1 + \overline{i2} + i3) \cdot (i1 + \overline{i2} + \overline{i3}) \cdot (\overline{i1} + \overline{i2} + \overline{i3})$
- g:
 - $g = \overline{i1} \cdot i2 \cdot \overline{i3} + \overline{i1} \cdot i2 \cdot i3 + i1 \cdot \overline{i2} \cdot \overline{i3} + i1 \cdot \overline{i2} \cdot i3 + i1 \cdot i2 \cdot \overline{i3}$
 - $g = (i1 + i2 + i3) \cdot (i1 + i2 + \overline{i3}) \cdot (\overline{i1} + \overline{i2} + \overline{i3})$

Es 3

Applicare il metodo delle mappe di Karnaugh ad ogni formula ottenuta per le varie uscite per minimizzare il numero di porte logiche.

A: $\overline{i3} \cdot \overline{i1} + i2 + i3 \cdot i1$

i1, i2 i3	00	01	11	10
0	1	1	1	0
1	0	1	1	1

B: $\overline{i1} + i3 \cdot i2 + \overline{i3} \cdot \overline{i2}$

i1, i2 i3	00	01	11	10
0	1	1	0	1
1	1	1	1	0

C: $i1 + i3 + \overline{i2}$

i1, i2 i3	00	01	11	10
0	1	0	1	1
1	1	1	1	1

D: $\overline{i3} * \overline{i1} + \overline{i1} * i2 + i3 * i1 * \overline{i2} + \overline{i3} * i2$

i1, i2 i3	00	01	11	10
0	1	1	1	0
1	0	1	0	1

E: $\overline{i3} * \overline{i1} + i2 * \overline{i3}$

i1, i2 i3	00	01	11	10
0	1	1	1	0
1	0	0	0	0

F: $\overline{i3} * \overline{i2} + i1 * \overline{i2} + i1 * \overline{i3}$

i1, i2 i3	00	01	11	10
0	1	0	1	1
1	0	0	0	1

G: $\overline{i3} * i2 + i1 * \overline{i2} + i1 * \overline{i3}$

i1, i2 i3	00	01	11	10
0	0	1	1	1
1	0	1	0	1

Es 4:

Disegnare il circuito risultante (solo AND, OR e NOT) che mette in relazione i 3 ingressi e le 7 uscite

