



software



a cura di Paolo Galassetti

## Simulatore digitale

di Pier Paolo Maccione - La Spezia

Sono molto contento di poter trovare in edicola una rivista di microcomputer come la Vostra, questo perché non soltanto riuscite a rendere semplici, nelle spiegazioni, anche le cose più difficili ma riuscite ad essere un punto d'incontro e di riferimento per tutti gli appassionati del settore da Voi trattato.

Detto questo, passo a presentarVi il programma che Vi invio.

Si tratta di un simulatore digitale che simula sulla HP 34C il comportamento delle principali porte logiche o operatori logici.

Il programma simula il comportamento delle porte logiche o operatori logici più comuni come AND, OR, NOT, NAND, NOR e anche OR esclusivo e NOR esclusivo. Gli ingressi sono uno per la porta NOT, due per le porte OR ex e NOR ex, da due a otto per le porte AND, NAND, OR, NOR. Il numero degli ingressi è scelto automaticamente dal numero di ingressi impostati, per esempio impostando 3 dati si avranno 3 ingressi. Perciò il numero di ingressi è uguale al numero di dati introdotti; per questo è importante impostare tutti gli ingressi, anche quelli nulli. I dati impostati devono essere in forma binaria cioè come 0 o 1. Perciò ogni

ingresso potrà avere solamente uno stato 1 o 0. Dopo aver impostato tutti gli ingressi desiderati si seleziona da programma il tipo di porta voluto e si ottiene l'uscita logica (0 o 1) di tale porta in corrispondenza degli ingressi impostati.

### Procedura

a) Premere il tasto A per cancellare le memorie

b) Impostare i valori degli ingressi: I<sub>1</sub> premere B..... 1 si ottiene sul visore il numero dell'ingresso appena impostato I<sub>2</sub> premere B.... 2

..... dopo avere impostato tutti gli ingressi voluti si va a c)

c) Selezionare la porta voluta, la corrispondenza tra la porta e il tasto da premere è data sotto:

GSB 1 AND  
GSB 2 OR  
GSB 3 NAND  
GSB 4 NOR  
GSB 5 NOT PER IMPOSTARE IL VALORE DELL'INGRESSO  
GSB 6 OR ex DELLA PORTA NOT BASTA VISUALIZZARLO

GSB 7 NOR ex SENZA ESEGUIRE IL PASSO b)

Supponiamo di dover studiare il comportamento di una porta NOR a tre ingressi che valgono: I<sub>1</sub>=0, I<sub>2</sub>=1, I<sub>3</sub>=0. Bisogna impostare i valori degli ingressi nel seguente modo: A (serve a cancellare le memorie), 0B.... 1, 1B..... 2, 0B..... 3; il numero dopo i puntini indica sempre il numero d'ordine dell'ingresso appena impostato. Dopo aver inserito i valori degli ingressi si procede alla selezione della porta, in questo caso NOR; c'è una corrispondenza tra le varie porte e le relative label dei sottoprogrammi, nel nostro caso bisognerà premere GSB 4 che corrisponde proprio alla porta NOR; dopo alcuni secondi di elaborazione apparirà il livello logico dell'uscita, nel nostro caso apparirà 0.

Supponiamo ora di voler utilizzare una porta NOT. La procedura è molto semplice: si imposta il valore dell'ingresso e si fa partire il tratto di programma interessato. Se in ingresso c'è un 1 la procedura è: 1 GSB 5, e si ottiene alla fine dell'elaborazione uno 0.

Esistono in commercio dei simulatori digitali che hanno lo scopo di mostrare in pratica il funzionamento delle logiche AND, NAND, OR, eccetera; tali simulatori sono indicati senz'altro come aiuto didattico per chi si accinge a studiare i fondamenti dell'elettronica digitale. Il programma proposto dal signor Maccione permette di trasformare la vostra 34C in un valido simulatore digitale. Il procedimento utilizza 9 registri dati più il registro I per l'indirizzamento indiretto; il suo funzionamento non dà problemi purché si stia attenti ad impostare i dati in modo corretto, perché il programma non prevede il controllo dei dati introdotti. Il passo 40 GTO 5 può essere eliminato perché non ha senso, essendo la LBL 5 situata immediatamente appresso. L'autore del programma non fa cenno, nella sua descrizione, al funzionamento delle varie routine.

I dati introdotti vengono memorizzati, dall'istruzione 12 STO (i), nei registri da 1 a 8; ciascuna cifra 0 o 1 seguita dalla pressione del tasto B viene memorizzata in uno di questi registri, dopodiché viene automaticamente incrementato di una unità il contenuto del registro R0; al termine dell'introduzione dei dati, il registro R0 contiene il numero di ingressi impostati. Per la funzione logica "and" (LBL 1), chiamando n il contenuto di R0, la calcolatrice somma il contenuto dei primi n registri a partire da R1, se tale somma è uguale al contenuto di R0 significa che tutti gli n ingressi hanno valore 1, in tal caso viene richiamata la

### Simulatore digitale per HP 34C

01	h	LBL A	35	h	LBL 3
f		CLEAR REG			GSB 1
8					GTO 5
EEX			h		LBL 4
3					GSB 2
05		CHS			GTO 5
STO f i			40	h	LBL 5
CLX				g	x=0
h		RTN			GTO 8
10	h	LBL B			GTO 9
g		ISG	45	h	LBL 6
STO f (i)					RCL 1
RCL f i					RCL 2
h		INT		f	x≠y
15		STO 0			GTO 8
h		RTN	50		GTO 9
h		LBL 1		h	LBL 7
g		GSB 0			GSB 6
DSE					GTO 5
20		GTO 1		h	LBL 8
RCL 0			55		1
RCL 9				h	RTN
f		x=y		h	LBL 9
GTO 8					O
25		GTO 9		h	RTN
h		LBL 2	60	h	LBL 0
g		GSB 0			RCL f (i)
DSE					STO + 9
GTO 2					RCL f i
30		RCL 9		h	INT
f		1	65	h	STO f i
x≤y			66	h	RTN
GTO 8					
GTO 9					

### Simulatore digitale per HP 41C

01	LBL "IN"	30 "FOR"	59 X#Y?
02	CLA	31 LBL 02	60 GTO 00
03	CLRG	32 XEQ 00	61 GTO 09
04	8 E-3	33 DSE 10	62 LBL "EXNOR"
05	STO 10	34 GTO 02	63 SF 05
06	CLX	35 RCL 09	64 "EXNOR"
07	CF 05	36 1	65 GTO 03
08	RTN	37 X<Y?	66 LBL 08
09	LBL "DATA"	38 GTO 08	67 FSTC 05
10	ARCL X	39 GTO 09	68 GTO 09
11	AVIEW	40 LBL "NAND"	69 "1."
12	ISG 10	41 SF 05	70 AVIEW
13	STO IND 10	42 "NAND"	71 RTN
14	RCL 10	43 GTO 01	72 LBL 09
15	INT	44 LBL "NOR"	73 FSTC 05
16	STO 00	45 SF 05	74 GTO 08
17	RTN	46 "NOR"	75 "1.0."
18	LBL "AND"	47 GTO 02	76 AVIEW
19	"AND"	48 LBL "NOT"	77 RTN
20	LBL 01	49 "NOT"	78 LBL 00
21	XEQ 00	50 RCL 01	79 RCL IND 10
22	DSE 10	51 X=0?	80 ST+ 09
23	GTO 01	52 GTO 08	81 RCL 10
24	RCL 00	53 GTO 09	82 INT
25	RCL 09	54 LBL "EXOR"	83 STO 10
26	X#Y?	55 "EXOR"	84 RTN
27	GTO 08	56 LBL 03	85 END
28	GTO 09	57 RCL 01	
29	LBL "OR"	58 RCL 02	

routine LBL 8 che provvede a visualizzare la cifra "1"; nel caso in cui non tutti gli ingressi si trovassero in condizione "1", la somma dei valori dei loro stati sarà minore di n e quindi verrà richiamata la LBL 9 a visualizzare "0". La funzione logica "OR" (LBL 2) lavora in modo analogo alla precedente ma, anziché confrontare la somma degli ingressi con il valore di R0, questa viene confrontata con lo 0, test quest'ultimo che risulterà positivo solo se tutte le entrate hanno livello logico 0. La funzione "OR ex" (LBL 6) confronta il contenuto dei registri R1 e R2; se sono uguali, in uscita avremo 0 altrimenti avremo 1. Le

funzioni "NAND", "NOR", e "NOR ex" sono ottenute dalle stesse "AND", "OR", e "OR ex" seguite però dalla routine LBL 5 (NOT) che inverte l'uscita (dà 0 al posto di 1 e viceversa).

Per chi non ha la 34C ma vuole usare il programma sulla sua 41C, ho provveduto a tradurre il programma del sig. Maccione nel linguaggio della 41C. Il programma si usa nello stesso modo dell'originale, ma con la differenza che i dati via via impostati, l'operatore scelto e il livello logico dell'uscita sono rappresentati tutti sul display in un'unica stringa; in verità tale stringa è

un po' "serrata", ma ho preferito far entrare tutto nel display, senza scorrimenti, piuttosto che correre il rischio di qualche errore di lettura dovuto allo "scrolling" dei caratteri più a sinistra. La routine "A" del programma originale corrisponde alla routine "IN" della versione per 41C e la routine "B" corrisponde alla routine "DATA". In questa versione del programma, contrariamente alla versione originale, la funzione "NOT" va richiesta come le altre (XEQ "NOT" oppure premere il tasto al quale la LBL è stata assegnata), dopo l'impostazione del dato tramite la routine "DATA".

## HP 41: qualche considerazione su.....

### ... Istruzioni di salto

Il principio fondamentale che anima questa rubrica è lo scambio delle idee e delle esperienze di programmazione. Senza necessariamente doverlo fare portando come esempio programmi interi, in questo angolino vorrei suggerire alcune soluzioni e idee che possono risultare utili a più d'uno nella compilazione di un programma.

Noto, in molti programmi inviati dai lettori, delle istruzioni di salto alfanumeriche, per esempio GTO "ABCD", che servono a far saltare il puntatore a etichette ALPHA poste nello stesso programma; ciò non sempre è una buona soluzione, perché un GTO ALPHA costringe il puntatore a cercare l'etichetta in tutta la memoria di programma con notevole spreco di tempo, mentre una istruzione GTO numerica

avrebbe dato luogo a un salto istantaneo alla label voluta. Inoltre, l'istruzione GTO numerica occupa soltanto due o tre byte di memoria (a seconda dell'indirizzo ad essa associato: 2 byte per gli indirizzi da 0 a 15 e 3 byte per gli indirizzi da 16 a 99), mentre un GTO ALPHA occupa dai 3 ai 9 byte.

Se la label alla quale si vuole inviare il puntatore è una label necessariamente alfanumerica, per esempio quella che dà il nome al programma come nell'esempio riportato in figura 1, può risultare utile aggiungere appresso ad essa una label numerica e, quando necessario, indirizzare i salti a quest'ultima come illustrato nell'esempio di fig. 2. Questa soluzione, anche se apparentemente impiega una istruzione in più, consente di risparmiare nel nostro caso ben 7 byte di memoria, oltre ad accorciare

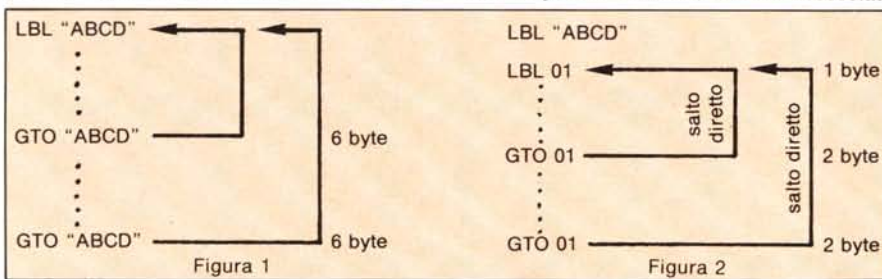
sensibilmente il tempo necessario per l'esecuzione del salto.

### ... Visualizzazione e stampa

Grossa confusione regna anche tra le varie istruzioni usate per arrestare i programmi contemporaneamente alla visualizzazione o stampa di messaggi, per esempio per l'input o l'output di dati. Senza dilungarmi in spiegazioni, pubblichiamo a fondo pagina uno specchio che dovrebbe servire da guida per la scelta dell'istruzione più adatta alle nostre necessità; ciascuna riga indica l'operazione eseguita da una istruzione (o coppia di istruzioni) in tre possibili condizioni di funzionamento.

In genere, la soluzione che io preferisco è di usare un PROMPT per le richieste di input, e un AVIEW o VIEW con il Flag 21 acceso per gli output; così facendo, il programma risulta perfettamente compatibile nell'uso con e senza stampante. Affinché il Flag 21 risulti sicuramente acceso durante lo svolgimento del programma, è bene collocare una istruzione SF 21 nella routine di inizializzazione del programma; tale istruzione provvederà ad accendere il Flag 21 ogni volta che viene inizializzato il programma, ciò è necessario poiché il Flag in questione si spegne ogni volta che la 41 viene accesa con la stampante scollegata.

P.G.



	SENZA STAMPANTE FLAG 21 SPENTO	SENZA STAMPANTE FLAG 21 ACCESO	CON STAMPANTE SU "MAN", FLAG 21 ACCESO
VIEW O AVIEW	visualizza il messaggio o risultato sul display ma non arresta l'esecuzione	visualizza il messaggio o risultato sul display e arresta l'esecuzione	stampa il messaggio o risultato, lo visualizza sul display ma non arresta l'esecuzione
VIEW O AVIEW Seguito da uno o più PSE	visualizza il messaggio o risultato sul display e sospende temporaneamente l'esecuzione (circa 1 sec. ogni "PSE")	—————	stampa il messaggio o risultato, lo visualizza sul display e sospende temporaneamente l'esecuzione
VIEW O AVIEW seguito da STOP	visualizza il messaggio o risultato sul display e arresta l'esecuzione	—————	stampa il messaggio o risultato, lo visualizza sul display e arresta l'esecuzione
PROMPT	arresta l'esecuzione e visualizza il contenuto del registro "ALPHA"	arresta l'esecuzione e visualizza il contenuto del registro "ALPHA"	arresta l'esecuzione e visualizza il contenuto del registro "ALPHA" (se la stampante è nel modo "NORMAL", tale contenuto viene anche stampato)
AON seguito da STOP	pone la macchina nel modo "ALPHA" e arresta l'esecuzione visualizzandone il contenuto	pone la macchina nel modo "ALPHA" e arresta l'esecuzione visualizzandone il contenuto	pone la macchina nel modo "ALPHA" e arresta l'esecuzione visualizzandone il contenuto
STOP	arresta l'esecuzione e visualizza il contenuto del registro X	arresta l'esecuzione e visualizza il contenuto del registro X	arresta l'esecuzione e visualizza il contenuto del registro X