

software COMMODORE 64

Il programma che vi presentiamo questo mese potrà essere utile a chi si occupa di progettazione di circuiti logici e, pubblicandolo, pensiamo di far cosa gradita soprattutto agli studenti degli Istituti Tecnici per i quali riveste un sicuro interesse didattico. Si tratta di un programma che, partendo da una rete

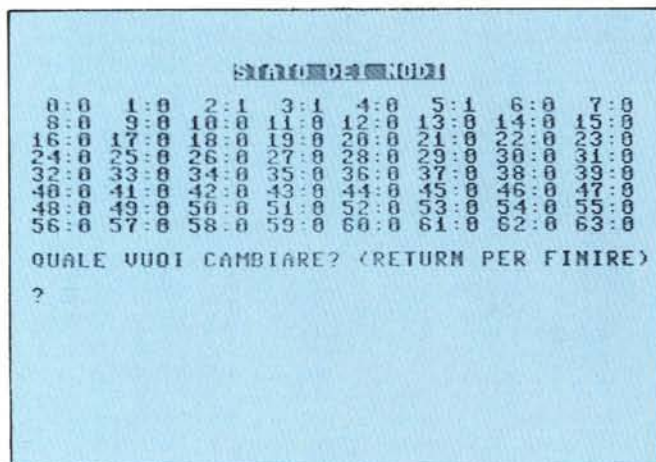
logica composta da And, Or, Flip-Flop ecc., la studia visualizzando dinamicamente sullo schermo il grafico degli stati in relazione allo scorrere del tempo e alle modifiche degli ingressi e delle uscite introdotte agendo sui generatori di costante simulabili da programma. Vale la pena dargli un'occhiata!

Simulatore di reti logiche di Alessandro Bedarida - Livorno

Oggigiorno è molto semplice verificare il corretto funzionamento di una rete logica per mezzo delle basette di montaggio sperimentale, le cosiddette «breadboard». Esiste tuttavia un metodo ancora più rapido ed economico, che consiste nella simulazione al calcolatore della rete in esame; il programma che vi proponiamo è appunto un simulatore di reti logiche.

Modalità d'uso

Dato il Run, il programma provvede alle inizializzazioni e alla lettura dal disco dei dati relativi ai simboli delle porte logiche. Dopo pochi istanti compare una schermata di presenta-

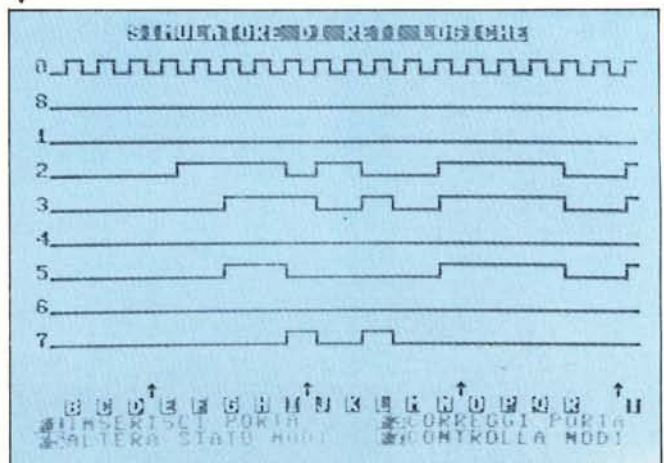
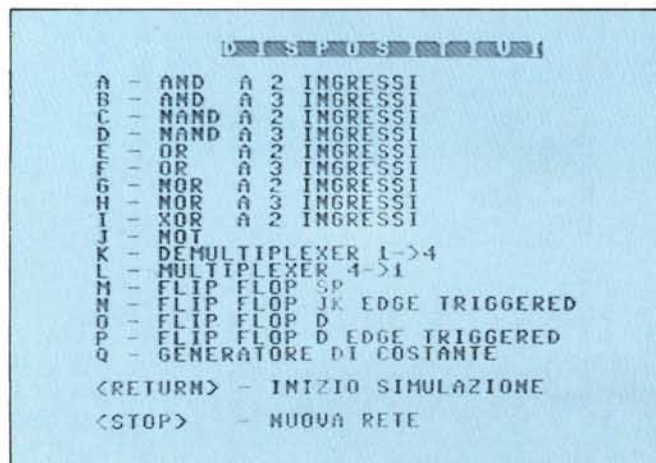


A sinistra un esempio di diagramma degli stati visualizzato per mezzo del programma.

In basso a sinistra la schermata attraverso la quale vengono selezionate le varie parti che compongono il circuito.

Da questa sezione si può forzare, in maniera coerente, la condizione di più nodi del circuito.

Questo programma è disponibile su disco e cassetta presso la redazione. Vedere l'elenco dei programmi disponibili e le istruzioni per l'acquisto a pag. 130.



2

```

180 DATA FLIP FLOP D,2,2
181 DATA "D",D,CLK,CLOCK,"D,CLK,Q,"D,CLOCK,"
182 DATA FLIP FLOP D EDGE TRIGGERED,2,2
183 DATA "D",D,CLK,CLOCK,"D,CLK,Q,"D,CLOCK,"
184 DATA GENERATORE DI COSTANTE,1,1
185 DATA "D",D,CLK,CLOCK,"D,CLK,Q,"D,CLOCK,"
186 PRINT "D",D,CLK,CLOCK,"D,CLK,Q,"D,CLOCK,"
194 GET AS:IF AS="" THEN 194
195 POKE VIC+21,0:POKE VIC+23,2:POKE VIC+29,2:POKE VIC,28:POKE VIC+1,52
196 GOSUB 7000
200 GOSUB 1000:IF ASC(AS)=3 THEN 1:REM INPUT DATI
210 GOSUB 2000:REM INPUT NODI DA VISUALIZZARE
220 GOSUB 2200:REM INIZIAL.SCHERMO
230 GOTO 250
240 GOSUB 2000:REM ELABORA
250 GOSUB 2100:REM VISUALIZZA
260 GET AS:FL=(AS=""):IF FL AND (PEEK(653)=1) THEN FL=0:GOTO 240
265 IF FL THEN 240
270 IF AS="" THEN 210
275 IF AS="" THEN 500
280 IF AS="" THEN GOSUB 2400:GOTO 220
285 IF AS="" THEN 200
290 IF AS="A" OR AS="T" AND AS<"*" OR AS="|" THEN 260
300 AS=ASC(AS)-64:IF A>128 THEN A=A-128
310 GOSUB 400:FL=-1:GOTO 240
400 I%(A)=NOT I%(A)
410 B=CO+879+2*A:POKE B,7-(PEEK(B) AND 15)
420 RETURN
500 PRINT "D",D,CLK,CLOCK,"D,CLK,Q,"D,CLOCK,"
501 FOR I=1 TO NT:NN=N*(D%(I,0)):IF D%(I,0)<11 THEN 509
502 IF D%(I,0)=11 THEN NN="DEMUX"
503 IF D%(I,0)=12 THEN NN="MUX"
504 IF D%(I,0)=13 THEN NN="FF-SR"
505 IF D%(I,0)=14 THEN NN="FF-JK"
506 IF D%(I,0)=15 THEN NN="FF-D"
507 IF D%(I,0)=16 THEN NN="FF-DE"
508 IF D%(I,0)=17 THEN NN="G.COS"
509 PRINT RIGHT$(STR$(I),2);": ";NN$;:NEXT I:PRINT
510 PRINT "D",D,CLK,CLOCK,"D,CLK,Q,"D,CLOCK,"
515 PRINT "(RETURN PER FIRE)":AS="":INPUT AS
520 IF AS="" THEN 220
530 T=VAL(AS):IF T=0 OR T>NT THEN 500
540 A=D%(T,0)
550 GOSUB 5000
560 GOTO 500
1000 REM INPUT DATI
1005 POKE VIC+21,1
1010 PRINT "D",D,CLK,CLOCK,"D,CLK,Q,"D,CLOCK,"
1015 FF=0
1020 PRINT "A - AND A 2 INGRESSI"
1025 PRINT "C - NAND A 2 INGRESSI"
1030 PRINT "E - OR A 2 INGRESSI"
1040 PRINT "G - NOR A 2 INGRESSI"
1045 PRINT "I - XOR A 2 INGRESSI"
1050 PRINT "K - DEMULTIPLEXER 1->4"
1060 PRINT "M - FLIP FLOP SR"
1062 PRINT "N - FLIP FLOP JK EDGE TRIGGERED"
1065 PRINT "O - FLIP FLOP D"
1070 PRINT "P - FLIP FLOP D EDGE TRIGGERED"
1075 PRINT "Q - GENERATORE DI COSTANTE"
1080 PRINT "R (RETURN) - INIZIO SIMULAZIONE"
1090 PRINT "S (STOP) - NUOVA RETE"
1100 GET AS:IF AS="" THEN 1100

```

(continua a pagina 122)

1

```

RETI LOGICHE
0 IF NOT A THEN A=-1:LOAD"RETI.DAT",8,1
1 POKE 55,127:POKE 56,135:CLR
2 REM *****
3 REM * SIMULATORE DI RETI LOGICHE *
4 REM * ALESSANDRO BEDARIDA *
5 REM * (C) 1985 *
6 REM * VERSIONE 2.1 *
7 REM *****
8 REM D%(I)=DISPOSITIVO N%(I)=STATO NODI V%(I)=GEN.DI COST.
9 REM SV%(I)=STATO PREC. NODI VISUALIZZ. NV=NUMERO NODI VISUALIZZATI
10 DATA 169,41,141,81,3,141,84,3,169,140,141,82,3,141,85,3,169,32,160,38
20 DATA 190,41,140,155,4,140,136,208,246,173,81,3,201,73,240,28,24
30 DATA 185,40,141,81,3,141,84,3,144,254,238,182,3,238,85,3,76,76,3,76
40 DATA 238,127,3,169,3,208,236,160,216,140,127,3,96
50 DATA 169,0,76,239,246,120,169,144,141,40,3,169,3,141,41,3,88,86
60 SC=35840:CO=5256:POKE 648,SC/256:REM BASE SCHERMO
61 POKE SC-1016,303:POKE 56576,(PEEK(56576) AND 252) OR 1:REM SPOSTA AREA VIC
62 REM INIZIALIZZAZIONE VIC
63 VIC=53248:POKE VIC+24,(PEEK(VIC+24) AND 15) OR 48:REM SPOSTA MEMORIA VIDEO
64 POKE VIC+32,1:POKE VIC+33,3:PRINT "D",D,CLK,CLOCK,"D,CLK,Q,"D,CLOCK,"
65 POKE VIC+39,1:POKE VIC+40,1:REM COLORE
66 POKE VIC+160:POKE VIC+1,129:REM X-Y SPRITE #0
67 POKE VIC+2,160:POKE VIC+3,80:REM X-Y SPRITE #1
68 POKE VIC+16,81:REM MSB X SPRITE #0 & #1
69 POKE VIC+63,3:POKE VIC+29,3:REM ESPANSIONE X & Y
70 GOSUB 6000
80 FOR I=0 TO 101:READ A:POKE 828+I,A:NEXT:SYS 917
100 ND=64:REM NUMERO MAX DISPOSITIVI
110 NV=63:REM NUMERO MAX NODI
120 IO=7:REM NUMERO MAX INGRESSI-USCITE
130 DIM D%(ND,IO),N%(NV),I%(20),N%(17),NO%(17),SC$(17),IO$(17,IO)
140 FOR I=1 TO 17:READ N%(I),NO%(I),SC$(I):FOR J=1 TO IO:READ IO$(I,J):NE
XT J,1
152 DATA AND,2,1
153 DATA "A",A,B,Y,,,,
154 DATA AND,3,1
155 DATA "A",A,B,C,Y,,,,
156 DATA NAND,2,1
157 DATA "A",A,B,Y,,,,
158 DATA NAND,3,1
159 DATA "A",A,B,C,Y,,,,
160 DATA OR,2,1
161 DATA "A",A,B,C,Y,,,,
162 DATA OR,3,1
163 DATA "A",A,B,C,Y,,,,
164 DATA NOR,2,1
165 DATA "A",A,B,Y,,,,
166 DATA NOR,3,1
167 DATA "A",A,B,C,Y,,,,
168 DATA XOR,2,1
169 DATA "A",A,B,Y,,,,
170 DATA NOT,1,1
171 DATA "A",A,Y,,,,
172 DATA DEMULTIPLEXER 1->4,3,4
173 DATA "A",A,B,C,D,Y,,,,
174 DATA MULTIPLEXER 4->1,6,1
175 DATA "A",A,B,C,D,Y,,,,
176 DATA FLIP FLOP SR,2,2
177 DATA "A",A,B,C,Y,,,,
178 DATA FLIP FLOP JK EDGE TRIGGERED,3,2
179 DATA "A",A,B,C,Y,,,,

```

```

1110 IF ASC(A$)=13 THEN FF=-1:POKE VIC+21,0:RETURN
1115 IF ASC(A$)=3 THEN RETURN
1120 IF A$="A" OR A$="Q" THEN 1100
1130 NT=NT+1:A=ASC(A$)-64:T=NT
1140 GOSUB 5000
1150 IF NT<ND THEN 1010
2000 REM ELABORAZIONE CICLO
2005 IF NOT FL THEN FOR I=1 TO NV:SVZ(I)=N:(VZ(I)):NEXT I:NZ(0)=NOT N:(0)
2010 FOR D=1 TO NT
2020 ON D<(D,0) GOSUB 3150,3250,3350,3450,3550,3650,3750,3850,3950,4050
2025 IF D<(D,0) THEN ON D<(D,0)-10 GOSUB 4150,4250,4350,4450,4550,4650,4750
2030 NEXT I:C1=C1+1+FL
2040 RETURN
2100 REM VISUALIZZAZIONE
2110 IF NOT FL THEN SYS 820
2120 FOR I=1 TO NV
2130 IF N:(VZ(I)) AND SVZ(I) THEN POKE SC+39+80*I,119
2140 IF N:(VZ(I)) AND NOT SVZ(I) THEN POKE SC+39+80*I,179
2150 IF NOT N:(VZ(I)) AND NOT SVZ(I) THEN POKE SC+39+80*I,111
2160 IF NOT N:(VZ(I)) AND SVZ(I) THEN POKE SC+39+80*I,176
2165 NEXT I
2180 IF C1-INT(C1/10)*10=0 THEN POKE SC+879,30
2190 RETURN
2200 REM INIZIALIZZAZIONE SCHERMO
2210 PRINT " "
2220 SYS 889
2230 FOR I=1 TO NV
2240 PRINT RIGHT$(STR$(VZ(I)),2):" "
2250 NEXT I
2260 PRINT " "
2265 PRINT " "
2267 PRINT " "
2270 FOR A=1 TO 20:IF I%(A) THEN GOSUB 410
2280 NEXT I
2290 RETURN
2300 REM INPUT NODI DA VISUALIZZARE
2310 PRINT " "
2320 PRINT " "
2330 NV=1
2340 VZ(NV)=-1:PRINT "M" NV:INPUT VZ(NV)
2350 IF VZ(NV)>=0 THEN 2380
2360 NV=NV+1:IF NV=0 THEN NV=1:VZ(1)=0
2370 RETURN
2380 IF NV<10 THEN NV=NV+1:GOTO2340
2390 RETURN
2400 REM CAMBIA NODI
2410 PRINT " "
2420 FOR I=0 TO NV STEP 8
2430 FOR J=1 TO I+7
2440 PRINT RIGHT$(STR$(J),2):";";CHR$(48-NZ(J)):"; "
2450 NEXT J
2460 NEXT I
2470 PRINT " "
2480 A=-1:INPUT A:IF A=-1 THEN RETURN
2490 IF A=0 AND A<=NV THEN NZ(A)=NOT N:(A)
2495 GOTO 2410
3100 REM AND 2
3150 NZ(DZ(D,3))=N:(DZ(D,1)) AND N:(DZ(D,2)):RETURN
3200 REM AND 3
3250 NZ(DZ(D,4))=N:(DZ(D,1)) AND N:(DZ(D,2)) AND N:(DZ(D,3)):RETURN
3300 REM AND 2
3350 NZ(DZ(D,3))=NOT(N:(DZ(D,1)) AND N:(DZ(D,2))):RETURN
3400 REM AND 3
3450 NZ(DZ(D,4))=NOT(N:(DZ(D,1)) AND N:(DZ(D,2)) AND N:(DZ(D,3))):RETURN
3500 REM OR 2
3550 NZ(DZ(D,3))=N:(DZ(D,1)) OR N:(DZ(D,2)):RETURN
3600 REM OR 3
3650 NZ(DZ(D,4))=N:(DZ(D,1)) OR N:(DZ(D,2)) OR N:(DZ(D,3)):RETURN

```

```

4480 NZ(DZ(D,4))=NOT N:(DZ(D,4)):N:(DZ(D,5))=NOT N:(DZ(D,5))
4490 DZ(D,6)=N:(DZ(D,2)):RETURN
4500 REM FF D
4550 IF N:(DZ(D,2)) THEN N:(DZ(D,3))=N:(DZ(D,1)):N:(DZ(D,4))=NOT N:(DZ(D,1))
4600 RETURN
4650 IF N:(DZ(D,2)) AND NOT D:(5) THEN N:(DZ(D,3))=N:(DZ(D,1)):N:(DZ(D,4))=NOT N:(DZ(D,1))
4660 DZ(D,5)=N:(DZ(D,2))
4670 RETURN
4700 REM GENERATORE DI COSTANTE
4750 NZ(DZ(D,2))=1:(DZ(D,1)):RETURN
5000 REM DEFINIZIONE INGRESSI E USCITE
5010 PRINT " "
5015 POKE SC+1017,30+POKE VIC+21,PEEK(VIC+21) OR 2
5017 PRINT " "
5020 PRINT " "
5022 PRINT:PRINT
5025 IF A<17 THEN 5030
5026 PRINT I0$(17,1):";":IF FF THEN PRINT " "
5027 INPUT D$:IF D$<"A" OR D$>"T" THEN 5000
5028 DZ(T,1)=ASC(D$)-64
5029 GOTO 5040
5030 FOR J=1 TO NIZ(A):PRINT I0$(A,J)
5032 IF FF THEN PRINT " "
5035 INPUT DZ(T,J):NEXT J
5040 PRINT " "
5045 PRINT:PRINT
5050 FOR J=NIZ(A)+1 TO NIZ(A)+NDZ(A):PRINT I0$(A,J)
5052 IF FF THEN PRINT " "
5055 INPUT DZ(T,J):NEXT J
5060 DZ(T,0)=A
5065 POKE VIC+21,PEEK(VIC+21) AND 253
5070 RETURN
6000 REM PRESENTAZIONE
6010 PRINT " "
6020 FOR I=5 TO 1 STEP -1
6030 PRINT " "
6040 PRINT " "
6050 PRINT " "
6060 PRINT " "
6070 PRINT " "
6080 PRINT " "
6090 PRINT " "
6100 POKE VIC+21,1
6110 RETURN
7000 REM STAMPA ISTRUZIONI
7010 PRINT " "
7020 PRINT " "
7030 PRINT " "
7040 PRINT " "
7050 PRINT " "
7060 PRINT " "
7070 PRINT " "
7080 PRINT " "
7090 PRINT " "
7100 PRINT " "
7110 PRINT " "
7120 PRINT " "
7130 PRINT " "
7140 PRINT " "
7150 PRINT " "
7160 PRINT " "
7200 GET A$:IF A$="" THEN 7200
7210 RETURN
10000 AD=2049:FOR I=1 TO 289:READ A$:NEXT I
10010 PU=PEEK(AD)+256*PEEK(AD+1):CS=0:FOR I=AD+4 TO PU-1:CS=CS+PEEK(I):NEXT I:CS=CS AND 255
10020 NL=PEEK(AD+2)+256*PEEK(AD+3):READ CC

```

Errata corrige

Per una svista tipografica, nel listato del programma GESBIB 64 pubblicato nel software C 64 di novembre sono state omesse le linee che pubblichiamo di seguito.

```

810 open4,dv,4,no$:ifdv=8thensys51829
811 Print#4,ts:Print#4,sl:Print#4,su
812 fori=0tosu-1:for n=0to5
  
```

zione seguita da una pagina che riassume le modalità d'uso ed i comandi principali. A questo punto è necessario svolgere un piccolo lavoretto «offline», cioè sulla carta: prendiamo lo schema della rete da simulare e numeriamo tutti i nodi (intendendo per nodo qualsiasi terminale d'ingresso o d'uscita presente nel circuito) a partire da 1 rispettando, per quanto possibile, un ordine crescente nel procedere dai nodi di ingresso verso quelli di uscita (ciò evidentemente non è possibile nel

caso si abbia a che fare con sistemi reazionati come, ad esempio, un flip-flop realizzato con elementi discreti). Il nodo 0 ha una funzione particolare che verrà descritta in seguito.

Quando ogni nodo è stato numerato, premendo un tasto qualsiasi viene mostrata una pagina che ci indica come inserire ciascuno dei 19 dispositivi possibili. Questi comprendono le porte logiche principali (AND, NAND, OR, NOR a 2 e 3 ingressi, XOR, NOT) ed alcuni dispositivi più com-

AND			NAND			OR			NOR			XOR			NOT	
A	B	Y	A	B	Y	A	B	Y	A	B	Y	A	B	Y	A	Y
0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	0	1
0	1	0	0	1	0	0	1	1	0	1	0	0	1	0	1	0
1	0	0	1	0	0	1	0	1	1	0	0	1	0	1		
1	1	1	1	1	0	1	1	1	1	1	0	1	1	0		

DEMULTEPLEXER 1->4							MULTIPLXER 4->1						
CE	A1	A0	Y3	Y2	Y1	Y0	A1	A0	D3	D2	D1	D0	Y
0	0	0	0	0	0	0	0	0	-	-	-	0	0
1	0	0	0	0	0	0	0	1	-	-	-	1	1
1	1	0	0	0	0	0	1	0	-	-	-	0	0
1	1	1	0	0	0	0	1	1	0	0	0	0	1

FLIP FLOP SR		FLIP FLOP JK		FLIP FLOP D		FLIP FLOP D ET	
S	R	Q	Q̄	J	K	Q	Q̄
0	0	0	1	0	0	0	1
0	1	0	0	0	1	0	0
1	0	1	0	1	0	1	0
1	1	0	0	1	1	0	0

```

6
10030 IF CS<XCC THEN PRINT*ERRORE DI BATTITURA ALLA LINEA*:NL
10040 AD=PU:IF N<10040 THEN 10010
10050 DATA 119,222,155,100,91,14,31,155,3,135,248,180,100,117,244,223
10060 DATA 169,144,132,52,242,247,3,174,109,89,115,172,234,198
10070 DATA 194,50,92,49,33,50,68,127,33,128,68,225,33
10080 DATA 0,68,77,33,78,68,87,33,78,123,86,22,173,40,160
10090 DATA 86,131,193,63,200,49,200,119,47,116,235,50,116,135
10100 DATA 163,169,64,78,107,111,186,8,84,4,143,87,20,122,221
10110 DATA 142,16,255,62,182,27,12,186,2,27,135,138,283,129,175
10120 DATA 93,62,62,129,3,91,110,15,111,243,83,11,205,248,219,153,140
10130 DATA 213,23,84,15,66,129,140,113,114,59,202,180,198,78,98,133,157,142
10140 DATA 70,76,85,243,144,123,141,130,134,142,179
10150 DATA 148,103,85,235,235,201,177,114,25,130,142,247,203,176
10160 DATA 135,161,98,200,142,171,142,150,11,109,244,96,236
10170 DATA 235,145,24,27,112,212,201,213,218,34,194,35
10180 DATA 211,162,282,163,226,240,195,241,213,168,9,142,160,97
10190 DATA 50,242,39,142,169,132,20,65,142,201,180,185,162,106,218,200,108
10200 DATA 350,159,153,142,212,207,49,142,201,180,232,175,229,93,142,150
10210 DATA 103,152,223,177,114,242,175,229,47,108,232,181,144,23,239,97,128,126
10220 DATA 112,171,249,185,238,114,165,227,243,3,142,181,144,23,239,97,128,126
10230 DATA 103,164,52,204,193,202,105,249,76,60,22,142,150,34,143,211,222,200
10240 DATA 145,210,156,71,184,179,202,12,94,219,228,228,166,69,143,7,160,64
10250 DATA 18
READY.
  
```

```

4
3700 REM NOR 2
3750 N:(D,D,3)=NOT(N:(D,D,1)) OR N:(D,D,2)):RETURN
3800 REM NOR 3
3850 N:(D,D,4)=NOT(N:(D,D,1)) OR N:(D,D,2)) OR N:(D,D,3)):RETURN
3900 REM XOR
3950 N:(D,D,3)=(N:(D,D,1))OR N:(D,D,2))AND NOT(N:(D,D,1))AND N:(D,D,2)):RETURN
4000 REM NOT
4050 N:(D,D,2)=NOTN:(D,D,1)):RETURN
4100 REM DEMUX
4150 FOR E=4 TO 7:N:(D,D,E)=0:NEXT
4160 IF N:(D,D,3) THEN N:(D,D,4)=2*N:(D,D,2)):RETURN
4170 RETURN
4200 REM MUX
4250 N:(D,D,7)=N:(D,D,(3-N:(D,D,1)))-2*N:(D,D,2)):RETURN
4300 REM FF SR
4350 IFN:(D,D,1)ORN:(D,D,2) THENN:(D,D,3)=N:(D,D,1):N:(D,D,4)=N:(D,D,4)):RETURN
4360 RETURN
4400 REM FF JK
4450 IF NOT N:(D,D,2)) OR D:(D,6) OR NOT(N:(D,D,1)) OR N:(D,D,3)) THEN 4480
4460 IF N:(D,D,1) AND N:(D,D,3)) THEN 4480
4470 N:(D,D,4)=N:(D,D,1):N:(D,D,5)=N:(D,D,3)):GOTO 4490
  
```

plexi (multiplexer, demultiplexer, flip-flop di 4 tipi e 20 generatori di costante).

Premendo uno qualsiasi dei tasti compresi fra «A» e «Q» si potrà inserire nel circuito il dispositivo corrispondente, che verrà visualizzato con i suoi ingressi e le sue uscite; quando ci verrà richiesto un dato relativo ad uno di essi dovremo inserire il numero che nel nostro schema gli corrisponde. Se in questa fase ci si accorge troppo tardi di aver commesso un errore, si vada avanti nell'inserimento degli altri dispositivi in quanto sarà possibile correggerlo più tardi. Anche durante l'inserimento dei dispositivi si dovrebbe fare attenzione ad iniziare con quelli più vicini ai nodi di ingresso.

Una volta inserito l'ultimo dispositivo (si deve ricordare che il massimo numero di dispositivi è 64 ed il massimo numero di nodi è 63, sufficienti comunque a simulare qualunque rete di complessità normale), premendo il tasto «RETURN» possiamo specificare i numeri dei nodi (al massimo 10) il cui stato logico deve essere visualizzato istante per istante, terminando con un «RETURN» a vuoto.

Finalmente inizia la simulazione vera e propria con la visualizzazione dell'andamento degli stati logici dei nodi selezionati e dei 20 generatori di costante disponibili. Premendo brevemente uno dei tasti «SHIFT» si fa avanzare la simulazione di un «quanto temporale», per cui verranno visualizzate le modifiche conseguenti alla variazione dello stato logico di un nodo o di un generatore di costante.

A questo punto è necessario spiegare esattamente cosa intendiamo per generatore di costante. Si tratta di un dispositivo, contraddistinto da una lettera compresa fra «A» e «T», che permette di forzare lo stato logico (0 oppure 1) di un nodo durante la simulazione agendo su un tasto della tastiera che fungerà da interruttore.

Per la visualizzazione rapida degli stati logici in più istanti successivi è sufficiente tenere premuto lo «SHIFT» o lo «SHIFT LOCK»; le frecce che compaiono in basso ogni 10 posizioni servono per meglio visualizzare lo scorrere del tempo.

Da questo ambiente si può passare in altri 4 ambienti premendo i tasti funzione come spiegato dal memorandum nella parte inferiore dello schermo. Con «F1» si possono inserire nuovi dispositivi da aggiungere alla rete con le modalità già viste oppure si può cancellare la rete e ricominciare da capo premendo il tasto «STOP». Con «F3» si visualizzano, ed eventualmente si modificano, i dispositivi inse-

riti. Per modificare un ingresso o un'uscita si deve inserire il numero del nuovo nodo al posto di quello preesistente e premere «RETURN»; se invece il nodo è già quello giusto, è sufficiente premere «RETURN» a vuoto. I dispositivi non possono essere cancellati ma, per escluderne uno dalla simulazione, è possibile assegnargli numeri di nodi che non fanno parte della rete, oppure riutilizzarlo in un punto diverso della rete stessa. Per mezzo del tasto «F5» si visualizzano gli stati logici

di tutti i 64 nodi a disposizione, con la possibilità di modificarne uno o più prima di riprendere la simulazione. Il tasto «F7», infine, serve per ridefinire i nodi di cui si vuole visualizzare istantaneamente lo stato logico.

Abbiamo detto che il nodo 0 ha una funzione speciale: ebbene questo nodo cambia il suo stato ad ogni quanto temporale, quindi non è altro che un oscillatore a disposizione dell'utente, il quale può utilizzarlo come base dei tempi. In altre parole, sul nodo con-

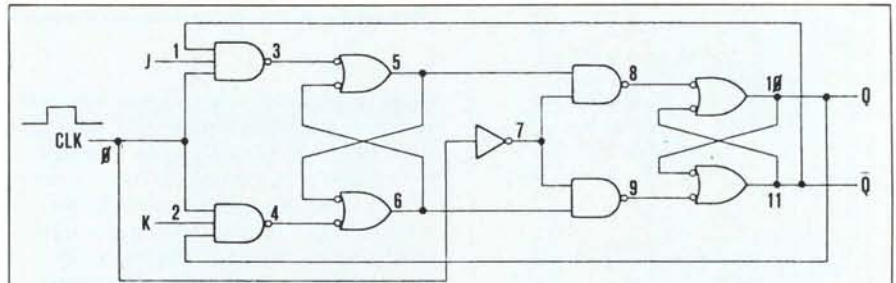


Figura 1

```

0 POKE55,127:POKE56,135:CLR
10 FOR I=34688 TO 34687+18*64
20 READ A
30 POKE I,A
40 NEXT I
50 INPUT "ASTRO O DISCO":W$
60 IF W$="N" THEN A$="1,1"
70 IF W$="D" THEN A$="8"
80 PRINT "PF43,127:PF44,135:PF45,2:PF46,140:S*:CHR*(34):*RET1.DAT*:CHR*(34):A$
90 POKE 198,1:POKE 631,13
100 END
9000 DATA48,15,254,128,12,6,128,12,6,128,12,6,251,239,254,10,44,6,10,44,6,10,32
9001 DATA0,250,47,128,2,40,0,2,40,0,2,40,0,3,239,190,0,8,8,255,136,8,193,136,8,1
93
9002 DATA136,8,255,224,8,192,96,8,192,96,8,255,224,8,0
9003 :
9010 DATA15,248,0,8,6,0,8,1,128,248,0,64,8,0,32,8,0,16,8,0,16,8,0,8,8,0,8,0,8
9011 DATA8,0,15,8,0,8,0,8,0,8,0,8,0,16,8,0,16,8,0,32,248,0,64,8,1,128,8,6,0,15
9012 DATA248,0,0
9013 :
9020 DATA15,248,0,8,6,0,8,1,128,248,0,64,8,0,32,8,0,16,8,0,16,8,0,8,8,0,8,0,8
9021 DATA248,0,15,8,0,8,0,8,0,8,0,8,0,16,8,0,16,8,0,32,248,0,64,8,1,128,8,6,0
9022 DATA15,248,0,0
9023 :
9030 DATA31,240,0,16,12,0,16,3,0,240,0,128,16,0,64,16,0,32,16,0,32,16,0,16,16,0
9031 DATA16,16,0,28,16,0,23,16,0,28,16,0,16,16,0,16,16,0,32,16,0,32,16,0,64,240
9032 DATA0,128,16,3,0,16,12,0,31,240,0,0
9033 :
9040 DATA31,240,0,16,12,0,16,3,0,240,0,128,16,0,64,16,0,32,16,0,32,16,0,16,16,0
9041 DATA16,16,0,28,240,0,23,16,0,28,16,0,16,16,0,16,16,0,32,16,0,32,16,0,64,240
9042 DATA0,128,16,3,0,16,12,0,31,240,0,0
9043 :
9050 DATA15,248,0,8,6,0,8,1,128,255,255,192,8,0,32,8,0,16,8,0,16,8,0,8,8,0,8,0,8
9051 DATA8,8,0,15,8,0,8,0,8,0,8,0,16,8,0,16,8,0,32,255,255,192,8,1,128,8,6
9052 DATA0,15,248,0,0
9053 :
9060 DATA15,248,0,8,6,0,8,1,128,255,255,192,8,0,32,8,0,16,8,0,16,8,0,8,8,0,8,0,8
9061 DATA8,255,255,255,8,0,8,0,8,0,8,0,16,8,0,16,8,0,32,255,255,192,8,1,128
9062 DATA8,6,0,15,248,0,0
9063 :
9070 DATA31,240,0,16,12,0,16,3,0,255,255,128,16,0,64,16,0,32,16,0,32,16,0,16,16
9071 DATA0,16,16,0,28,16,0,23,16,0,28,16,0,16,16,0,16,16,0,32,16,0,32,16,0,64,25
9072 DATA255,128,16,3,0,16,12,0,31,240,0,0
9073 :
9080 DATA31,240,0,16,12,0,16,3,0,255,255,128,16,0,64,16,0,32,16,0,32,16,0,16,16
9081 DATA0,16,16,0,28,255,255,247,16,0,28,16,0,16,16,0,16,16,0,32,16,0,32,16,0,0,4
9082 DATA255,255,128,16,3,0,16,12,0,31,240,0,0
9083 :

```

trassegnato dallo 0 è automaticamente presente un generatore di onda quadrata.

Esempi di utilizzazione

Vediamo come simulare il funzionamento del flip flop JK Master-slave di figura 1. Inseriamo innanzitutto i due generatori di costante necessaria per fornire al circuito gli ingressi J e K. Per fare ciò premiamo il tasto «Q» e, quando ci viene richiesto di contraddistinguere il generatore con una lettera compresa fra «A» e «T», rispondiamo «J» (la lettera non è comunque significativa nel senso che potevamo scegliere anche la lettera A o un'altra). L'uscita sarà invece il nodo 1. Ripetiamo l'operazione con un altro generatore (ad esempio K) ponendo questa volta l'uscita sul nodo 2. Dobbiamo ora inserire le due porte NAND a 3 ingressi premendo il tasto «D» e impostando come nodi d'ingresso i nodi 11, 1 e 0 (che è il generatore di clock) e come uscita il nodo 3; la seconda porta

NAND ha invece i nodi 0, 2 e 10 come ingressi e il 4 come uscita. A questo punto si presenta il problema di dover inserire due porte non disponibili, le OR con gli ingressi negati. Ricordando il teorema di De Morgan: (NOT A) OR (NOT B) = NOT (A AND B) notiamo che le porte mancanti possono essere sostituite con due NAND a 2 ingressi. Procediamo così finché non abbiamo inserito tutte le 9 porte che compongono il circuito e premiamo «RETURN». Scegliamo di visualizzare gli stati logici dei nodi 1, 0 e 2, che sono gli ingressi e 10 e 11, che sono le uscite. E adesso... via alla simulazione!

Particolarità del programma

Il programma, salvo la routine di scorrimento orizzontale della parte centrale dello schermo che è in linguaggio macchina, è scritto interamente in Basic. Per questo motivo la velocità di elaborazione, specialmente nel caso di reti complesse non è elevata; a

questo inconveniente si rimedia facilmente compilando il programma con il Petspeed.

I dati relativi agli sprite sono contenuti in un file di nome «RETI.DAT» che risiede su memoria di massa (disco o nastro) e che viene creato dal programma ausiliario «DATI SPRITE», preparato a parte.

Nel caso si voglia usare l'unità a nastro come memoria di massa, l'unica modifica da fare consiste nel cambiare il numero del dispositivo da 8 ad 1 nella linea 0 del programma principale.

Istruzioni per la battitura

1) Battere il programma DATI SPRITE e salvarlo su disco o su nastro.

2) Dare il comando «RUN 10000» e attendere circa un minuto e mezzo. Se il programma è stato inserito correttamente comparirà il solito messaggio «READY», altrimenti una routine di controllo indicherà il numero di riga nel quale è stato commesso un errore di battitura. Quando sono state eseguite le correzioni e non si hanno più segnalazioni di errore si può salvare la copia corretta del programma (eventualmente dopo aver eliminato le linee 10000 e seguenti, che ormai non servono più) e cancellare la copia precedente.

Nota - Come potrete notare dai listati riguardanti questi programmi, essi questa volta non vengono stampati con i numeri di checksum, come avviene di consueto, in quanto il controllo introdotto da A. Bedarida è molto efficiente.

3) Lanciare il programma corretto, il quale provvederà a scrivere su memoria di massa il file «RETI.DAT». Nella versione pubblicata, tale memoria è rappresentata da un dischetto ma, con semplici modifiche, l'operazione di salvataggio e successivo caricamento può essere indirizzata al nastro.

4) Battere il programma principale (RETI LOGICHE) e salvarlo.

5) Ripetere il punto 2 per il programma principale.

6) A questo punto il programma è pronto per essere eseguito. Se lo si vuole compilare con il Petspeed, poiché questo non ammette il dimensionamento dinamico degli array, si avrà un arresto durante la compilazione e la richiesta di dimensionamento statico degli array d%, n% e io\$; per far proseguire la compilazione si dovrà rispondere: dim d% (64,7), dim n% (63), dim io\$ (17,7).

```

9090 DATA 15,248,0,9,6,0,9,1,128,255,255,192,9,0,32,9,0,16,9,0,16,9,0,8,9,0,8,9,0
9091 DATA 8,9,0,15,9,0,8,9,0,8,9,0,8,9,0,16,9,0,16,9,0,32,255,255,192,9,1,128,9,6
9092 DATA 0,15,248,0,0
9093 :
9100 DATA 0,0,0,0,0,24,0,0,22,0,0,17,128,0,16,96,0,16,24,0,16,6,0,16,1,128,16,0
9101 DATA 112,240,0,95,16,0,112,16,1,128,16,6,0,16,24,0,16,96,0,17,128,0,22,0,0,2
4
9102 DATA 0,0,0,0,0,0,0,0
9103 :
9110 DATA 31,255,248,16,0,24,16,0,104,16,1,143,16,6,8,16,24,8,16,96,8,17,128,15,2
2
9111 DATA 0,8,248,0,8,22,0,8,17,128,15,16,96,8,16,24,8,16,6,8,16,1,143,16,0,104,1
6
9112 DATA 0,24,31,255,248,1,0,64,1,0,64,0
9113 :
9120 DATA 31,255,248,24,0,8,22,0,8,241,128,8,16,96,8,16,24,8,16,6,8,240,1,136,16
9121 DATA 0,104,16,0,31,16,0,104,240,1,136,16,6,8,16,24,8,16,96,8,241,128,8,22,0
9122 DATA 8,24,0,8,31,255,248,1,0,64,1,0,64,0
9123 :
9130 DATA 31,255,248,16,0,8,23,0,72,244,0,175,23,0,168,17,0,168,23,0,88,16,0,8,16
9131 DATA 0,8,16,0,8,16,0,8,16,0,8,16,0,232,16,0,8,23,0,72,21,0,168,23,0,168,246
9132 DATA 0,175,21,0,88,16,0,8,31,255,248,0
9133 :
9140 DATA 31,255,248,16,0,8,23,0,72,242,0,175,18,0,168,18,0,168,22,0,88,16,0,8,19
9141 DATA 0,8,20,0,8,244,0,8,20,0,8,19,0,232,16,0,8,21,0,72,21,0,168,22,0,168,245
9142 DATA 0,175,21,0,88,16,0,8,31,255,248,0
9143 :
9150 DATA 31,255,248,16,0,8,22,0,72,21,0,168,245,0,175,21,0,168,22,0,88,16,0,8,16
9151 DATA 0,8,16,0,8,16,0,8,16,0,8,16,0,232,16,0,8,19,0,72,20,0,168,244,0,175,20
9152 DATA 0,168,19,0,88,16,0,8,31,255,248,0
9153 :
9160 DATA 31,255,248,16,0,8,22,0,72,21,0,168,245,0,175,21,0,168,22,0,88,16,0,8,16
9161 DATA 0,8,16,0,8,16,0,8,16,0,8,16,0,232,16,0,8,24,0,72,20,0,168,242,0,175,20
9162 DATA 0,168,24,0,88,16,0,8,31,255,248,0
9163 :
9170 DATA 0,16,67,14,40,189,9,69,11,8,130,8,8,0,8,0,8,0,8,0,8,0,8,0,8,0,8,255,128
9171 DATA 8,0,8,62,0,8,34,0,8,62,0,1,8,0,2,8,0,4,8,0,8,0,8,127,0,127,73,0,73
9172 DATA 146,0,146,0
10000 AD=2049:FOR I=1 TO 1152:READ A$:NEXT
10010 PU=PEEK(AD)+256*PEEK(AD+1):CS=0:FOR I=AD+4 TO PU-1:CS=CS+PEEK(I):NEXT:CS=C
S AND 255
10020 NL=PEEK(AD+2)+256*PEEK(AD+3):READ CC
10030 IF CS<>CC THEN PRINT"ERRORE DI BATTITURA ALLA LINEA":NL
10040 AD=PU:IF NL<10090 THEN 10010
10050 DATA 158,194,232,109,235,67,121,25,53,209,128,27,115,75,58,216,4,217,58
10060 DATA 216,216,107,58,172,191,80,58,172,238,80,58,16,22,199,58,16,86,141
10070 DATA 58,194,33,188,58,194,50,132,58,24,31,199,58,202,34,147,58,79,73,97
10080 DATA 58,245,214,41,58,27,205,212,58,32,245,212,58,19,202,221,58,19,196
10090 DATA 217,58,39,193,157,188,34,143,211,225,117,78,148,58
10100 DATA 107
READY.
```