

DA RESTITUIRE INSIEME AGLI ELABORATI e A TUTTI I FOGLI  
 → NON USARE FOGLI NON TIMBRATI  
 → ANDARE IN BAGNO PRIMA DELL'INIZIO DELLA PROVA  
 → NO FOGLI PERSONALI, NO TELEFONI, SMARTPHONE/WATCH, ETC

COGNOME \_\_\_\_\_

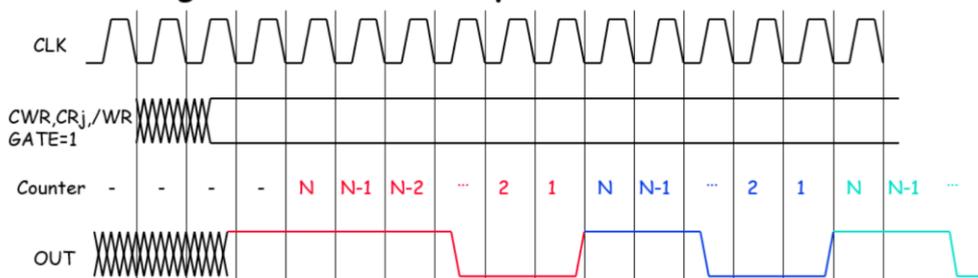
NOME \_\_\_\_\_

NOTE: I FOGLI UTILIZZATI PER RAGIONAMENTI VANNO RICONSEGNA TI ANCHE SE BIANCHI; PER I FILE ELETTRONICI:

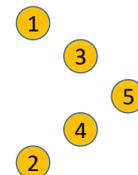
- per l'esercizio 3 consegnare un file di testo di nome <COGNOME>.txt
- per l'esercizio 4 consegnare DUE files: il file del programma VERILOG di nome <COGNOME>.v e il file del diagramma temporale (screenshot o copy/paste) <COGNOME>.png

3) [8/30] Spiegare con proprie parole il funzionamento del "Modo 3" del timer 8254, il cui diagramma temporale è riportato in figura. Inoltre, indicare con precisione: i) il significato dei segnali rappresentati in tale diagramma, ii) come deve essere impostata la parola di controllo CWR e il relativo registro di conteggio per ottenere questo diagramma supponendo di utilizzare N=21504 e il contatore n.2 in conteggio binario.

• **Modo 3: generatore d'onda quadra**



4) [22/30] Realizzare in Verilog una rete sequenziale secondo il modello di Mealy-Ritardato che accenda 5 led di una "freccia a destra" nel modo seguente: durante il primo ciclo tutte i led sono spenti; nel ciclo successivo si accendono i led 1 e 2, nel secondo ciclo si accendono i led 1,2,3,4; nel ciclo successivo tutti i 5 led sono accesi; poi la sequenza si ripete, cioè al ciclo successivo led tutti spenti, poi 1,2, poi 1,2,3,4 e così via. L'ingresso X su un bit è un interruttore generale che indica con X=1 che i led (governati dall'uscita Z su 3 bit) si accendono secondo la sequenza descritta, se X=0 tutti i led devono stare spenti. Tracciare il diagramma di temporizzazione come verifica della correttezza dell'unità riportando i segnali clock, /reset, ingresso X, uscita Z, registro di stato STAR.

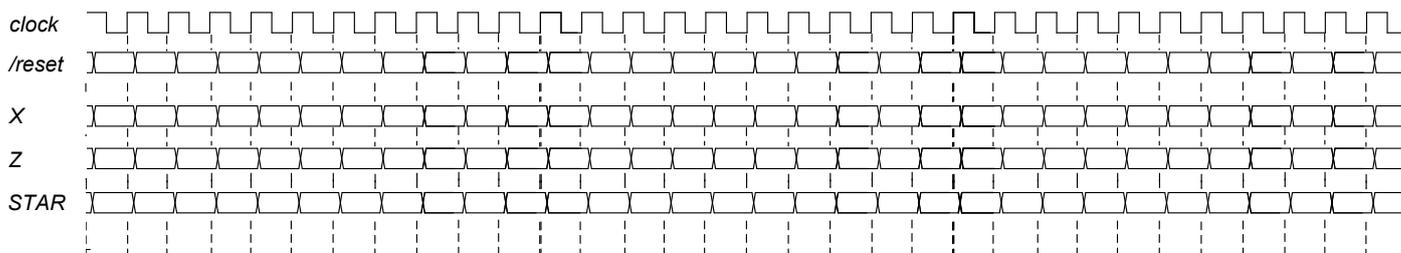


Utilizzare il testbench allegato, **tracciare il diagramma di temporizzazione [11 punti su 30]** come verifica della correttezza del modulo realizzato.

```

`timescale 100ms/1ms
module TopLevel;
reg reset_;initial begin reset_=0; #22 reset_=1; #300; $stop; end
reg clock ;initial clock =0; always #5 clock <=!clock);
reg x; initial begin x=0; #40 x=1; #260 $finish; end
wire[1:0] STAR = Xxx.STAR;
wire[2:0] z=Xxx.z;
XXX Xxx(x, z, clock, reset_);
Endmodule
    
```

Il diagramma temporale deve rappresentare i seguenti segnali:

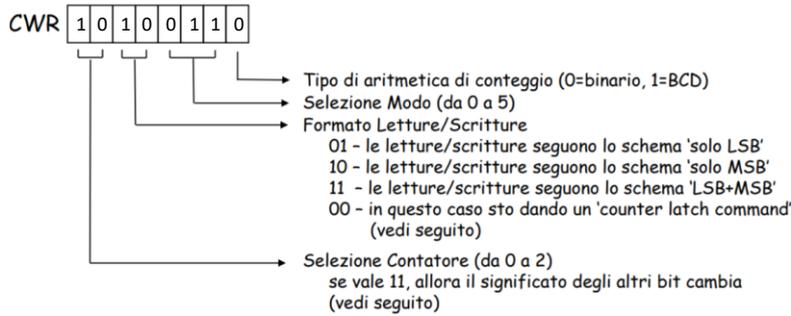


**ESERCIZIO 3**

Il modo 3 viene utilizzato per realizzare sull'uscita OUT (es. OUT2 per CR2) un'onda quadra con duty-cycle di circa il 50% alla frequenza determinata dal clock di riferimento; l'impulso viene generato dopo un ritardo temporale pari a  $N/f_c$  essendo N la costante di tempo scritta nel registro di conteggio (es. CR2), mentre  $f_c$  è la frequenza applicata sul piedino CLK corrispondente al contatore di interesse (es. CLK2 per CR2); il ritardo viene conteggiato a partire da un istante di innesco che parte dal fronte in discesa del clock successivo alla scrittura di CRW (mentre il valore "1" è applicato sul piedino GATE (es. GATE2 per CR2)).

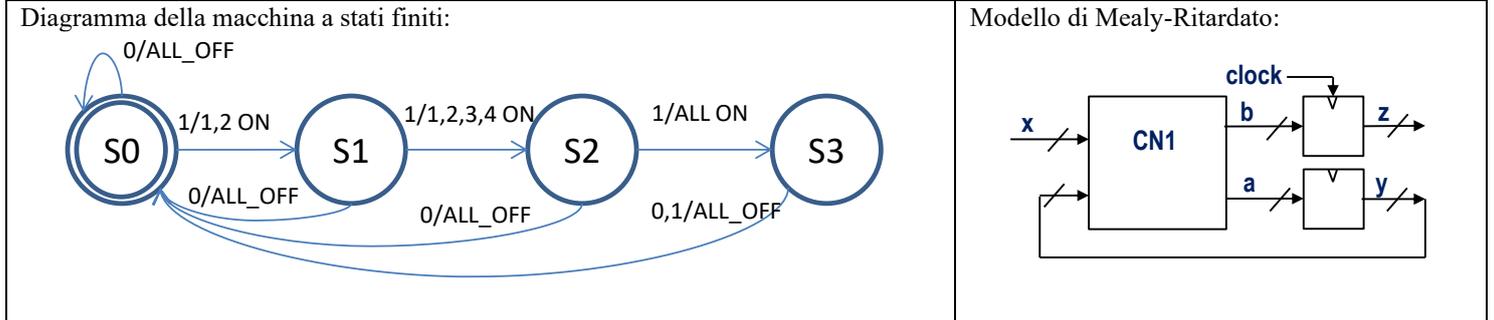
i) In figura sono rappresentati i segnali appena discussi (OUT, GATE, CLK); inoltre, "Counter" (ovvero CR2) indica il valore assunto dal contatore durante il conteggio, mentre CWR indica il valore impostato nel registro CWR e /WR è il segnale di scrittura applicato per poter scrivere nei registri CR2 e CWR.

Determinazione della parola di controllo CRW:



ii) La parola di controllo deve valere 1010'0110=0xA6, essendo possibile effettuare una sola scrittura da 8 bit per scrivere il byte più significativo (MSB) della costante  $N=21504=0x5400$  (che ha una dimensione di 16 bit), ovvero basta scrivere 0xA6 in CRW e 0x54 in CR2.

**ESERCIZIO 4**



Possibile codice Verilog del modulo da realizzare

```

module XXX(x,z,clock,reset_);
input clock,reset_;
input x;
output [2:0] z;
reg [1:0] STAR;
reg [2:0] OUTR;
parameter S0='B00,S1='B01,S2='B10,S3='B11;
assign z = OUTR;

always @(reset_==0) #1 begin STAR<=S0; OUTR<='B000; end
always @(posedge clock) if(reset_==1) #2
caseX (STAR)
    S0: begin STAR<=(x==0)?S0:S1; OUTR<=(x==0)?'B000:'B001; end
    S1: begin STAR<=(x==0)?S0:S2; OUTR<=(x==0)?'B000:'B011; end
    S2: begin STAR<=(x==0)?S0:S3; OUTR<=(x==0)?'B000:'B111; end
    S3: begin STAR<=(x==0)?S0:S0; OUTR<='B000; end
endcase
endmodule
    
```

Diagramma di Temporizzazione:

