

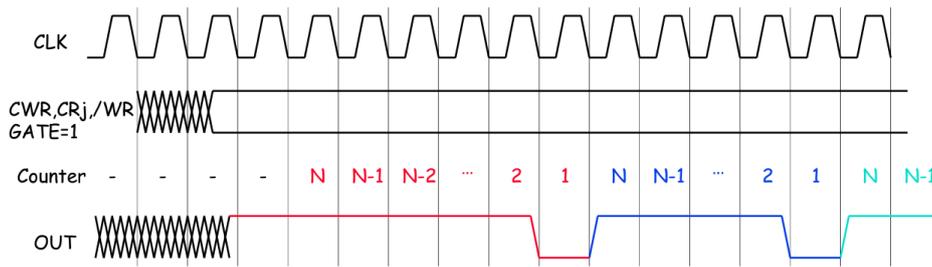
DA RESTITUIRE INSIEME AGLI ELABORATI e A TUTTI I FOGLI  
 → NON USARE FOGLI NON TIMBRATI  
 → ANDARE IN BAGNO PRIMA DELL'INIZIO DELLA PROVA  
 → NO FOGLI PERSONALI, NO TELEFONI, SMARTPHONE/WATCH, ETC

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

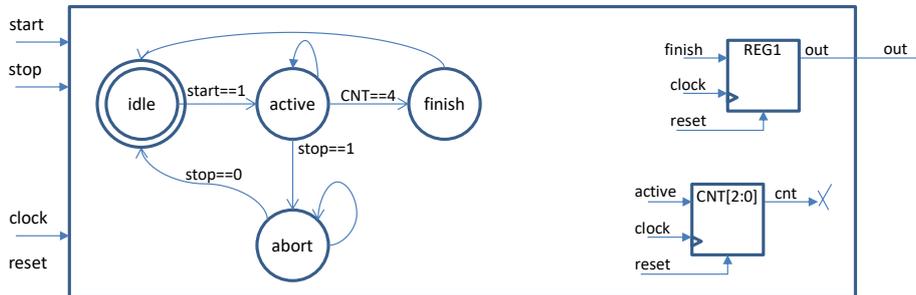
- per l'esercizio 3 consegnare un file di testo di nome <COGNOME>.txt
- per l'esercizio 4 consegnare DUE files: il file del programma VERILOG di nome <COGNOME>.v e il file del diagramma temporale (screenshot o copy/paste → usare tasto 'STAMP') <COGNOME>.png

3) [10/30] Spiegare con proprie parole il funzionamento del "Modo 2" 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=64, il contatore n.1 in conteggio binario.

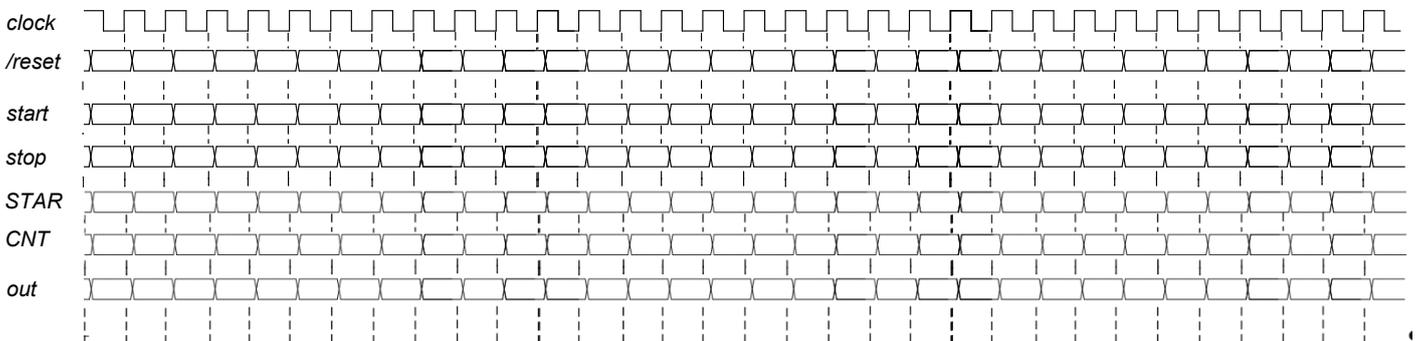
- **Modo 2: rate generator (divisore di frequenza)**  
 → genera un onda quadra con duty-cycle pari a (N-1)/N



4) [20/30] Descrivere e sintetizzare in Verilog una rete sequenziale utilizzando un modello appropriato (fra Moore, Mealy, Mealy-Ritardato) i cui ingressi e uscita sono descritti in figura; al suo interno la rete è descritta dal diagramma a stati della stessa figura e conterrà due registri: un contatore CNT che usa interi da 0 a 4 e un registro di uscita REG1. Gli stimoli di ingresso sono dati dal seguente modulo Verilog Testbench.



**Tracciare il diagramma di temporizzazione [8/20 punti]** come verifica della correttezza dell'unità. Nota: si può svolgere l'esercizio su carta oppure con ausilio del simulatore salvando una copia dell'output (diagramma temporale) e del programma Verilog su USB-drive del docente. Modello del diagramma temporale da tracciare:



```

module Testbench;
    reg reset; initial begin reset = 0; #7 reset = 1; #300; $stop; end
    reg clock; initial clock = 0; always #5 clock <= (!clock);
    reg start, stop;
    wire [1:0] STAR = Xxx.STAR;
    wire [2:0] CNT = Xxx.CNT;
    initial begin start <= 0; stop <= 0; wait(reset == 1);
        #20 start = 1; #10 start = 0; #100 start = 1; #10 start = 0; #20 stop = 1; #10 stop = 0; #50 start = 1; #200
        $finish;
    end
    XXX Xxx(start, stop, clock, reset_, out);
endmodule
    
```

**ESERCIZIO 3**

Il modo 2 viene utilizzato per realizzare sull'uscita OUT (es. OUT1 per CR1) un'onda quadra con duty-cycle pari a  $(N-1)/N$  e periodo pari a  $N/fc$  essendo N la costante di tempo scritta nel registro di conteggio (es. CR1), mentre fc è la frequenza applicata sul piedino CLK corrispondente al contatore di interesse (es. CLK1 per CR1).

i) In figura sono rappresentati i segnali appena discussi (OUT, GATE, CLK); inoltre, "Counter" (CR1) 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 CR1 e CWR.

ii) La parola di controllo deve valere 0101'0100=0x54, essendo necessario sufficiente una scrittura del byte più significativo (LSB) per scrivere i 16 bit della costante  $N=64=0x40$ , ovvero basta scrivere 0x54 in CRW e in CR1 (il byte più significativo si assume re-impostato automaticamente a zero).



**ESERCIZIO 4**

Codice Verilog del modulo da realizzare (possibile soluzione con Mealy-Ritardato):

```

module XXX(start,stop,clock,reset_, out);
    input    clock,reset_;
    input    start,stop;
    output   out;
    reg      REG1;
    reg[1:0] STAR;
    reg[2:0] CNT;
    parameter idle='B00,active='B01,finish='B10,abort='B11;
    always @(reset_==0) #1 begin STAR<=idle; REG1<=0; end
    assign out=REG1;

    always @(posedge clock) if(reset_==1) #3
    casex(STAR)
        idle:    begin STAR<=(start==1)?active:idle; CNT=0; REG1<=0; end
        active:  begin STAR<=(CNT==4)?finish:
                    (stop==1)?abort:active; CNT<=CNT+1; REG1<=0; end
        finish:  begin STAR<=idle; REG1<=1; end
        abort:   begin STAR<=(stop==0)?idle:abort; REG1<=0; end
    endcase
endmodule
    
```

**Diagramma di Temporizzazione:**

