

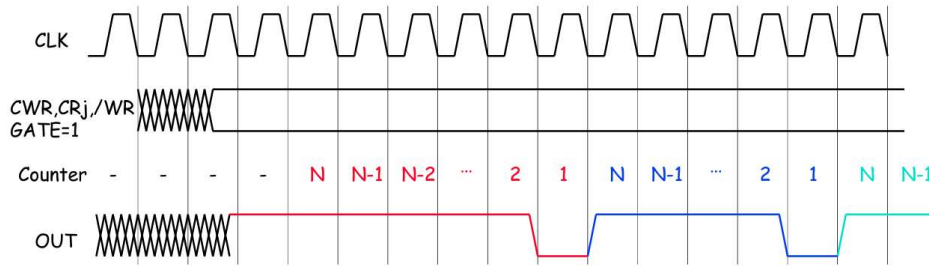
DA RESTITUIRE INSIEME AGLI ELABORATI e A TUTTI I FOGLI
 → NON USARE FOGLI NON TIMBRATI
 → ANDARE IN BAGNO PRIMA DELL'INIZIO DELLA PROVA
 → NO FOGLI PERSONALI, NO TELEFONI, SMARTPHONE/WATCH, ETC

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

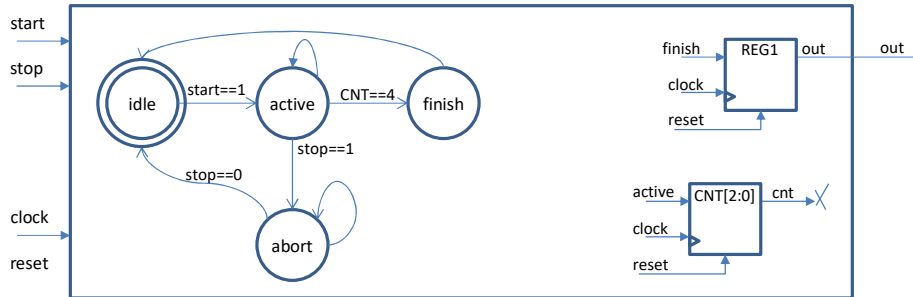
- per l'esercizio 3 consegnare un file di testo di nome <COGNOME>.txt
- per l'esercizio 4 consegnare DUE files: il file del programma VERILOG di nome <COGNOME>.v e il file del diagramma temporale (screenshot o copy/paste → usare tasto 'STAMP') <COGNOME>.png

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.

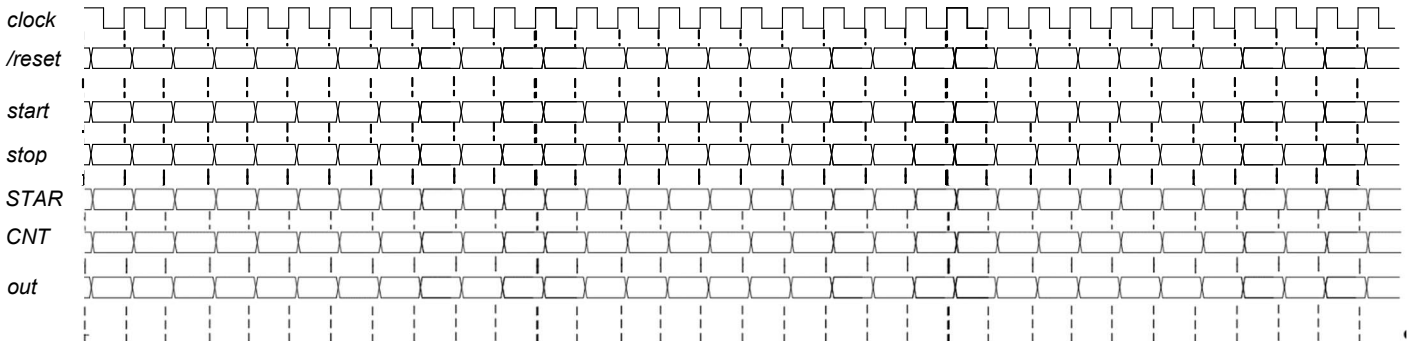
• **Modo 2: rate generator (divisore di frequenza)**
 → genera un onda quadra con duty-cycle pari a (N-1)/N



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
    
```