

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:

- 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) [12/30] Data una funzione booleana con 4 ingressi x_3, x_2, x_1, x_0 e una uscita z tale che l'uscita vale 1 qualora gli ingressi rappresentino in binario una delle cifre decimali 2, 3, 5, 7:

- [4/30] Utilizzando le mappe di Karnaugh, sintetizzare tale funzione in forma algebrica (suggerimento: gli ingressi sono 4 cifre booleane x_3, x_2, x_1, x_0 e valori 10, 11, 12, 13, 14, 15 sono non-specificati).
- [4/30] Realizzare la funzione del punto a utilizzando solo porte NAND
- [4/30] Assumendo 150ps per il ritardo di ogni porta NAND, calcolare il ritardo del cammino più critico e il ritardo del cammino più veloce.

4) [18/30] Realizzare in Verilog una macchina a stati di tipo Mealy-ritardato con un ingresso X a 1 bit e un'uscita Z a 1 bit, che riconosca esclusivamente sequenze non interallacciate del tipo 1, 0, 1, 0. All'inizio e dopo ogni riconoscimento, la macchina si resetta automaticamente. Al reset l'uscita Z è 0. L'uscita Z diventa 1 esattamente per un ciclo di clock al riconoscimento della sequenza. Tracciare il diagramma di temporizzazione [punti 8/30] corrispondente alla esecuzione del modulo Verilog realizzato, come verifica della correttezza dell'unità riportando i segnali clock, /reset, ingresso X , uscita Z e stato STAR per la durata complessiva. Nota: salvare una copia dell'output (diagramma temporale) e del programma Verilog su USB-drive del docente.

```
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;
  wire Z;
  wire [2:0] STAR=Xxx.STAR;
  initial begin X=0;
    wait(reset_==1);
    @(posedge clock); X<=1;@(posedge clock); X<=0;@(posedge clock); X<=1;
    @(posedge clock); X<=0;@(posedge clock); X<=1;@(posedge clock); X<=0;
    @(posedge clock); X<=1;@(posedge clock); X<=1;@(posedge clock); X<=0;
    @(posedge clock); X<=0;@(posedge clock); X<=1;@(posedge clock); X<=0;
    @(posedge clock); X<=1;@(posedge clock); X<=0;@(posedge clock); X<=1;
    @(posedge clock); X<=0;@(posedge clock); X<=1;@(posedge clock); X<=0;
    @(posedge clock); X<=0;@(posedge clock); X<=0;@(posedge clock); X<=0;
  end
  ricseq1010mealyrit Xxx(Z, X, clock, reset_);
endmodule
```