

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) [3/20] Fornendo una spiegazione ragionata, con il dettaglio del significato dei vari bit per il formato dell'istruzione, determinare a quale istruzione assembly RISC-V corrisponde la seguente stringa binaria (codice macchina)

0010 1100 1010 0110 1011 1001 1010 0011

ricordando che i codici operativi (opcode/funct3/funct7 in esadecimale) delle principali istruzioni viste sono:
 ADD: 33/0/00, LD: 03/3/imm, SD: 23/3/imm, BEQ: 63/0/(imm÷2), LUI: 37/imm[31:12], SLT: 33/2/0

- 4) [17/20] Descrivere e sintetizzare in Verilog una rete che realizzi un "Contatore a 3 bit" in grado di contare verso l'alto o verso il basso a seconda che il valore dell'ingresso X sia rispettivamente 0 o 1.

Il contatore usa internamente la codifica di Gray dello stato ovvero varia il suo stato interno varia secondo la sequenza 000, 001, 011, 010, 110, 111, 101, 100, mentre l'uscita Z (su 3 bit) rappresenta il valore corrispondente all'intero da 0 a 7 nell'ordine naturale (al codice Gray 011 corrisponde 2, a 010 corrisponde 3 e così via).

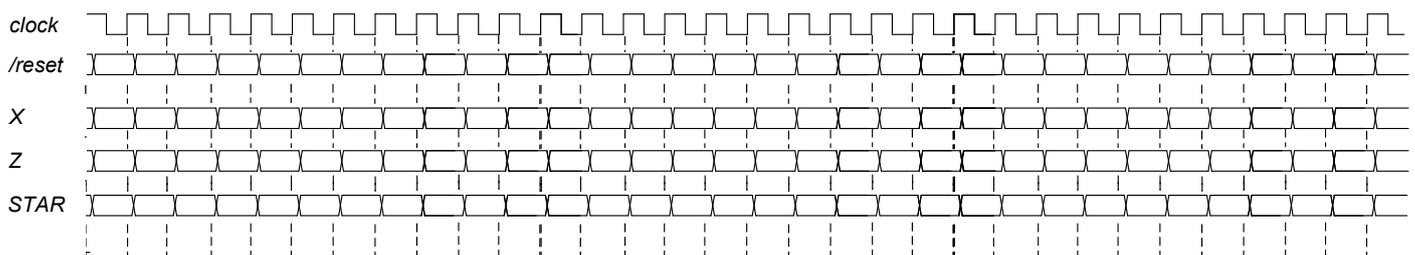
Utilizzando il testbench allegato, tracciare il diagramma di temporizzazione [9 punti su 17] come verifica della correttezza del modulo realizzato.

```

module Testbench;
  reg reset_; initial begin reset_=0; #22 reset_=1; end
  reg clock ;initial begin clock=0; forever #5 clock <=!clock); end
  reg X;
  wire[2:0] Z=ud3g.z;
  wire[2:0] STAR=ud3g.STAR;
  initial begin X=0; wait(reset_==1);
    #50 X=0; #20 X=1; #50 X=0; #20 X=1; #50 $finish;
  end
  UPDOWN3BITGRAY ud3g(X,clock,reset_, Z);
endmodule

```

Il diagramma temporale deve rappresentare i seguenti segnali:



SOLUZIONE

ESERCIZIO 3

Dai primi 7 bit vediamo che l'opcode e' 0b010 0011 ed un eventuale funct3 e' 3 ovvero opcode/funct3=0x23/0x3: questo corrisponde alla istruzione SD che ha un formato S nel RISC-V. Quindi i restanti bit possono essere raggruppati come segue:

0010 110	01010	01101	011	1 0011	010 0011
imm[11:5]	rs2	rs1	funct3	imm[4:0]	opcode

A questo punto si possono interpretare i campi rs1,rs2,imm che quindi corrispondono rispettivamente ai registri x10, x13 mentre l'immediato 0010 1101 0011 corrisponde all'esadecimale 0x2D3 ovvero al decimale 723. Quindi l'istruzione corrispondente è:
sd x10,723(x13)

ESERCIZIO 4

Codice Verilog del modulo da realizzare

```

module UPDOWN3BITGRAY(x,clock,reset_, z);
input clock,reset_;
input x;
output[2:0] z;
reg [2:0] STAR, OUTR;
parameter S0='B000,S1='B001,S2='B011,S3='B010,S4='B110,S5='B111,S6='B101,S7='B100;
assign z = OUTR;
always @(reset_==0) #1 STAR<=S0;
always @(posedge clock) if(reset_==1) #3
case(x)
S0: begin OUTR<='B000; STAR <= (x==0) ? S1:S7; end
S1: begin OUTR<='B001; STAR <= (x==0) ? S2:S0; end
S2: begin OUTR<='B010; STAR <= (x==0) ? S3:S1; end
S3: begin OUTR<='B011; STAR <= (x==0) ? S4:S2; end
S4: begin OUTR<='B100; STAR <= (x==0) ? S5:S3; end
S5: begin OUTR<='B101; STAR <= (x==0) ? S6:S4; end
S6: begin OUTR<='B110; STAR <= (x==0) ? S7:S5; end
S7: begin OUTR<='B111; STAR <= (x==0) ? S0:S6; end
endcase
endmodule
    
```

Diagramma di Temporizzazione:

