Seconda verifica AESO 2021-22

L'indirizzo email della persona che ha risposto (**null**) è stato registrato quando hai inviato questo modulo.

1. Email *

2. Una memoria "dinamica"

Contrassegna solo un ovale.

- ◯ E' più veloce di una memoria statica è più lento
- ◯ Richiede hardware aggiuntivo per il rinfresco del contenuto, che altrimenti verrebbe perso in breve lasso di tempo
- ◯ A parità di capacità è più costosa di una memoria statica è più economica

3. Un dispositivo di memoria "base", nello stesso ciclo di clock:

Contrassegna solo un ovale.

- ◯ Permette di leggere e scrivere una cella di indirizzo dato
- ◯ Permette di leggere o scrivere una cella di indirizzo dato
- ◯ Permette di leggere una cella ad un certo indirizzo e di scrivere una cella ad un altro indirizzo no se non è multiporta

4. In una memoria multiporta:

Contrassegna solo un ovale.

- ◯ Posso effettuare operazioni su indirizzi diversi nello stesso ciclo di clock
- ◯ Posso effettuare esclusivamente letture da indirizzi diversi nello stesso ciclo di clock
- ◯ Posso effettuare esclusivamente scritture su indirizzi diversi nello stesso ciclo di clock

si possono avere + indirizzi diversi che comandano sia letture che scritture

5. In quale tipo di memoria è richiesto un numero maggiore di componenti (porte logiche, transistor) per implementare un singolo bit:

Contrassegna solo un ovale.

- ( ) RAM dinamica
- (X) RAM statica richiede + transistors e porte logiche x ciascun bit
- ( ) ROM

6. Si consideri una memoria modulare interallacciata che utilizza 4 componenti di memoria standard da 1G parole per implementare una memoria da 4G parole, e si indichi quale delle seguenti affermazioni sono vere:

Contrassegna solo un ovale.

- ( ) Due parole di indirizzi consecutivi si trovano nello stesso componente di memoria da 1G (mai)
- (X) Due parole di indirizzi consecutivi si trovano nei componenti di memoria da 1G di indice i e (i+1)%4
- ( ) e' sempre possibile leggere 4 parole consecutive in un singolo ciclo di clock, purchè possa essere bypassato il multiplexer che normalmente sceglie l'unica parola determinata dall'intero indirizzo a 32 bit

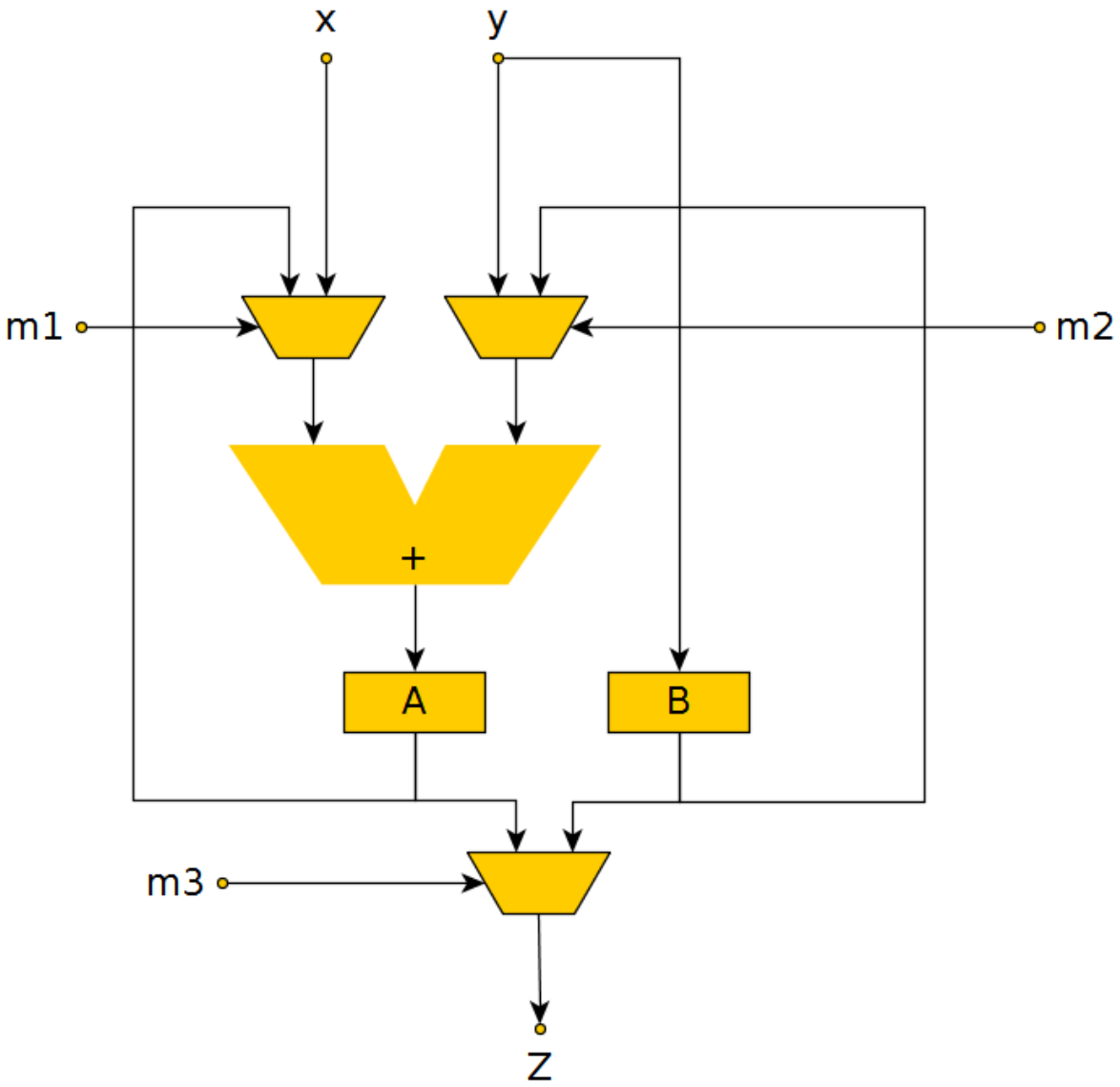
è possibile solo se gli indirizzi cominciano con un indirizzo che ha tutti 0 alla fine [x x x|00] $\log_2$(# moduli)

7. Un componente PLA (Programmable logic array):

Contrassegna solo un ovale.

- (X) può implementare solo reti combinatorie
- ( ) può implementare solo reti sequenziali
- ( ) può implementare sia reti combinatorie che sequenziali

8. La rete sequenziale in figura (sono evidenziati gli ingressi e le uscite, A e B sono registri con segnale di abilitazione sempre a 1) è



Contrassegna solo un ovale.

- Una rete di Moore (l'uscita deriva da registri, che sono stato interno)
- Una rete di Moore (perchè l'uscita non dipende da X e Y)
- Una rete di Mealy (perchè l'uscita è calcolata a partire dallo stato interno e dall'ingresso m3)
- Una rete di Mealy (perchè il contenuto dei registri, da cui si prende l'uscita) è determinato dagli ingressi X e Y)

x e y determinano il contenuto dei registri di stato (solamente)

9. Se due reti sequenziali, una di Moore e una di Mealy, risolvono lo stesso problema:

Contrassegna solo un ovale.

- ( ) La rete di Moore utilizza un numero di stati maggiore rispetto a quelli utilizzati dalla rete di Mealy ≥
- (X) La rete di Moore può richiedere un numero di stati maggiore rispetto a quelli utilizzati dalla rete di Mealy
- ( ) La rete di Moore utilizza un numero di stati minore rispetto a quelli utilizzati dalla rete di Mealy mai
- ( ) La rete di Moore può richiedere un numero di stati minore rispetto a quelli utilizzati dalla rete di Mealy
- ( ) La rete di Moore utilizza un lo stesso numero di stati rispetto a quelli utilizzati dalla rete di Mealy può, ma può anche richiederne di più

10. Una modulo PRIMITIVE in Verilog

Contrassegna solo un ovale.

- ( ) definsce una rete combinatoria con un numero qualunque di ingressi, ciascuno di un numero di bit qualunque, e una singola uscita, di un numero di bit qualunque falso
- (X) definsce una rete combinatoria con un numero qualunque di ingressi, ciascuno di un numero di bit qualunque, e una singola uscita, di un solo bit
- ( ) definisce una rete combinatoria di un ingresso con un numero di bit pari a una potenza di due, e una uscita di un solo bit falso
- ( ) definisce una rete combinatoria qualunque falso

11. Il seguente modulo Verilog

```
module x(output [7:0]z,
         output      c,
         input [7:0] a,
         input [7:0] b,
         input       o);

  assign
        {c,z} = (o == 1'b0 ? a+b : a-b);

endmodule
```

genera 9 bit

Contrassegna solo un ovale.

- implementa una alu che implementa somma e sottrazione di numeri interi a 8 bit
- implementa una alu che implementa somma e sottrazione di numeri interi a 8 bit e genera un flag pari a 1 se c'è stato riporto
- potrebbe implementare una alu che fa somma e sottrazione di numeri interi a 8 bit, e che genera un flag pari a 1 ma andrebbe scritta una assign separata per il bit c, perchè l'unica assign presente non è corretta

{a,b}=... con a e b di m e n bit; b prende i primi n bit (→ significativi) e a gli altri m

12. Come potremmo completare il modulo che segue in modo che calcoli la parità di 3 bit?

```
module par3(output p, input a, input b, input c);

  wire p1;

  assign
    p1 = (a&&b) || (~a && ~b);

  ...

endmodule
```

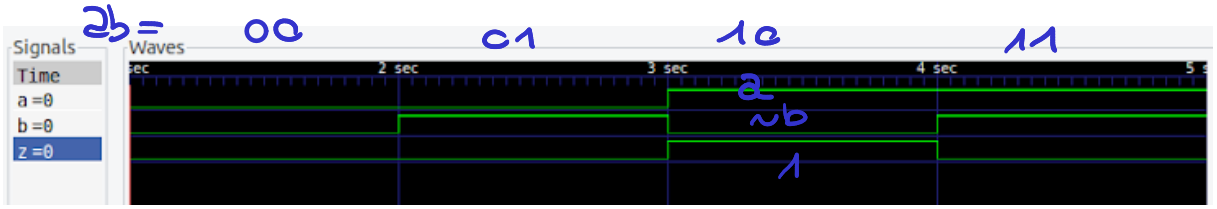
Contrassegna solo un ovale.

- assign p = (p1&&c || ~p1&&~c);
- assign p = (p1&&~c || ~p1&&c);
- assign p = (p1&&c || p1&&~c);

il primo assign calcola la parità dei primi due bit

la parità dei tre bit è 1 se p1=1 e c=0 (pari + "0" = pari) oppure p1=0 e c=1 (dispari + "1" = pari)

13. Quale dei moduli, genera la seguente traccia di uscita:



Contrassegna solo un ovale.

- module p(output z, input a, input b); assign z = a && b; endmodule
- module p(output z, input a, input b); assign z = a && ~b; endmodule
- module p(output z, input a, input b); assign z = a || b; endmodule
- module p(output z, input a, input b); assign z = a || ~b; endmodule

14. Quanto valgono a e b dopo l'esecuzione del codice seguente?

```
reg a;
reg b;

...

a = 0;
b = 1;

# 1
  a <= b;
b <= a;
```

Contrassegna solo un ovale.

- entrambi 1
- entrambi 0
- a 1 e b 0
- a 0 e b 1
- il valore di emtrambi è indeterminato

15. Cosa calcola il modulo c4?

```
module c1(output z, input a, input b);
   assign z = a&&b || ~a&&~b;
endmodule // comp

module c4(output z, input [3:0]a, input [3:0]b);

   wire [3:0] r;

   genvar i;
   generate
      for(i=0; i<4; i=i+1)
           c1 myC(r[i], a[i], b[i]);
   endgenerate

   assign z = &r;

endmodule // c4
```

a b	z
0 0	1
0 1	0
1 0	0
1 1	1

4 moduli c1 calcolano 1 bit ciascuno = "1" sse i due bit sono uguali

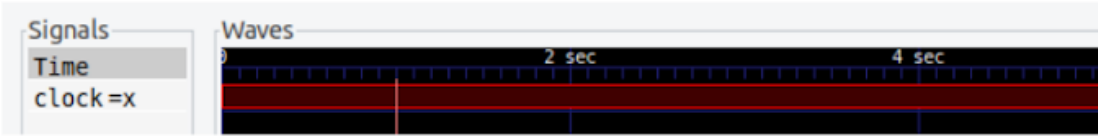
z = &r calcola AND (&) bit a bit ⇒ 1 sse tutti i bit sono "1"

Contrassegna solo un ovale.

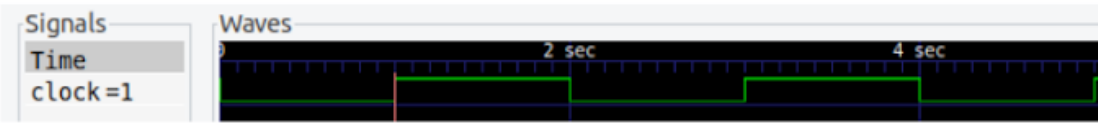
- z = 1 se e solo se a > b
- ~~z = 1 se e solo se a = b~~
- z = 1 se e solo se a < b
- non è un modulo valido, perchè la sintassi utilizzata per il modulo c1 è sbagliata
- non è un modulo valido, perchè la sintassi utilizzata per il modulo c4 è sbagliata

16. Si consideri il codice "reg clock; always #1 clock = ~clock; " e la serie di tracce seguenti, e si marchino le affermazioni vere

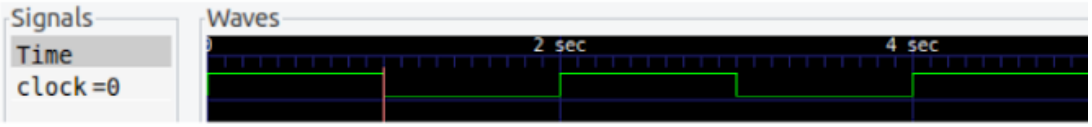
A)



B)



C)



Seleziona tutte le voci applicabili.

- [ ] la traccia risultante è la traccia B) se l'unico assegnamento al registro clock nel blocco "initial" è uno statement "#2 clock = 0" ← no, altrimenti avrei visto 'x' per #2
- [ ] la traccia risultante è la traccia B) se l'unico assegnamento al registro clock nel blocco "initial" è uno statement "#2 clock <= 0" ← idem
- [x] la traccia risultante è la la traccia A) se non si inizializza il registro clock in un blocco "initial"
- [ ] la traccia risultante è la traccia C) se l'unico assegnamento al registro clock nel blocco "initial" è uno statement "#2 clock = 1" ← idem

Se non assegno un valore iniziale ← vero
↓
clock è v "x" ⇒ "x" (quindi "x" per sempre)

17. Si hanno un file mod.v e un file test.v che contengono rispettivamente il codice che definisce una rete combinatoria (mod.v) ed il relativo testbench (test.v), che contiene una coppia di comandi "$dumpfile("test.vcd");" e "$dumpvars;". Quale delle seguenti serie di comandi permette di visualizzare la traccia di esecuzione?

Contrassegna solo un ovale.

manca modulo ↙

- ( ) iverilog test.v; a.out; gtkwave test.vcd
- (x) iverilog test.v mod.v; a.out; gtkwave test.vcd
- ( ) iverilog test.v mod.v; gtkwave test.vcd

↑ manca esecuzione

vanno completati test e def modulo, eseguito il risultato e visualizzato la traccia