



Laboratorio di Architettura degli Elaboratori I
Corso di laurea triennale in Informatica
Università degli Studi di Milano

Appello del 21 Febbraio 2020

Indicazioni generali

- L'esame dura 2 ore ed è composto da 3 esercizi.
- È proibito l'accesso ad Internet, tranne per <https://upload.di.unimi.it> e <http://teaching.basilico.di.unimi.it/> unicamente dal PC dell'aula.
- Ogni device che consenta di comunicare va riposto fuori dalla postazione di lavoro.
- È possibile prendere appunti e consultare materiale di studio.
- Vanno consegnati unicamente i files richiesti da ciascun esercizio. I files vanno nominati esattamente come indicato.
- Verranno corretti solo gli esercizi che sono stati consegnati correttamente e che non generano errori.

Svolgimento dell'esame

- Collegarsi a <https://upload.di.unimi.it> e loggarsi, usando le credenziali UNIMI, nella sessione "Laboratorio Architettura degli Elaboratori I (Appello 21 Febbraio 2020)"
- Usando la web form, uploadare i files richiesti da ogni esercizio.
- È possibile caricare un file con lo stesso nome più volte, il sistema mantiene tutte le versioni caricate. A meno di indicazioni esplicite da parte dello studente, verranno corrette le ultime versioni caricate.
- Può succedere che la sessione di upload scada causando l'impossibilità di caricare files. In tal caso, basta ripetere la procedura di login.

Esercizio 1

Upload: `esercizio1.circ`

Sia data una rete combinatoria con tre ingressi e due uscite. I tre ingressi A, B, C della rete sono usati per definire un numero binario su tre bit. Le due uscite f_1 ed f_2 forniscono le seguenti informazioni sul numero ABC:

- $f_1 = 1$ se e solo se ABC rappresenta un numero maggiore o uguale a 5;
- $f_2 = 1$ se e solo se ABC non è un numero primo (0 non è primo).

(A: 2 pts) Realizzare il sotto-circuito chiamato `sop1` che implementi f_1 come SOP non minimizzata (un prodotto per ogni mintermine).

(B: 2 pts) Realizzare il sotto-circuito `sop2` che implementi f_2 come SOP non minimizzata (un prodotto per ogni mintermine).

(C: 3 pts) Semplificare i due circuiti precedenti, riducendo il numero di porte di ogni circuito. Si salvi il risultato nel sotto-circuito `semplificazione`.

(D: 3 pts) Implementare, nel sotto-circuito `solo-nand` i due circuiti del punto C usando solo porte NAND (definire i sotto-circuiti per AND, OR, NOT realizzati con solo NAND) e unirli in un solo circuito in cui siano presenti entrambe le uscite f_1 ed f_2 .

Esercizio 2

Upload: `esercizio2.circ`

Un contatore ciclico modulo 6, una volta raggiunto il valore 5, ricomincia a contare da 0. Lo stato del contatore è il suo valore c . Nella transizione da stato corrente a stato prossimo, il valore passa da c a $c + 1 \pmod{6}$. Sia n il minor numero di bit per rappresentare lo stato del contatore modulo 6.

(A: 3 pts) Si implementi in un sotto-circuito chiamato `next_mod6` la rete combinatoria che, dati in input gli n bit dello stato corrente (il valore c), calcoli in uscita gli n bit dello stato prossimo (il valore $c + 1 \pmod{6}$).

(B: 3 pts) Si utilizzi il sotto-circuito `next_mod6` per implementare, in un altro sotto-circuito `mod_6`, un contatore ciclico modulo 6 così definito:

- input: il clock (1 bit) e il segnale $+$ (1 bit); sul fronte di salita del clock, se $+$ è pari a 1 il contatore passa allo stato prossimo; in ogni altro caso il contatore mantiene il suo valore corrente;
- output: n bit che rappresentano lo stato corrente del contatore (il valore c) e 1 bit di parità che vale 1 se e solo se c è pari (0 è pari.)

(C: 4 pts) Si utilizzi il sotto-circuito `mod_6` per implementare un banco di 4 contatori modulo 6 definito come segue:

- input: il clock (1 bit), un segnale $+$ (1 bit) e un indirizzo (2 bit); sul fronte di salita del clock, se $+$ è pari a 1 il contatore identificato dai 2 bit di indirizzo passa allo stato prossimo;
- output: il valore (n bit) e la parità (1 bit) del contatore modulo 6 indirizzato.

Esercizio 3

Upload: `esercizio3.circ`

Si implementi un sotto-circuito chiamato `estrai-slt` che riceva in input un segnale di `clock` (1 bit), un segnale `estrai` (1 bit) e un segnale `slt` (1 bit).

(A: 3 pts) Sul fronte di salita del clock, se il segnale `estrai` è pari ad 1 il circuito estrae 2 numeri binari casuali ciascuno su 8 bit. Al suo interno, il circuito memorizza gli ultimi due numeri estratti, che indichiamo con a e b .

(B: 3 pts) Sul fronte di salita del clock, se il segnale `slt` è pari a 1 il circuito controlla se a è strettamente minore di b .

(C: 3 pts) In caso positivo, il circuito incrementa un contatore interno su 8 bit (inizialmente posto a 0). Il valore di tale contatore costituisce l'uscita del circuito.

(D: 1 pts) Sul fronte di salita del clock, se entrambi i segnali `estrai` e `slt` sono pari a 1, il circuito si comporta come nel caso A. In tutti gli altri casi il circuito mantiene stato e uscita invariati.