

Laboratorio di Architettura degli Elaboratori

A.A. 2015/16 — Circuiti Logici

Per ogni lezione, sintetizzare i circuiti combinatori o sequenziali che soddisfino le specifiche date e quindi implementarli e testarne il comportamento mediante il programma simulatore Logisim.

Lezione 1

Mediante le mappe di Karnaugh, sintetizzare i circuiti *combinatori* che soddisfino le seguenti specifiche:

1.1 Esercizio

Progettare un circuito combinatorio, con tre segnali di input, che calcola la “maggioranza” di essi: il circuito fornisce in uscita 1 se almeno due ingressi sono 1, fornisce 0 altrimenti.

1.2 Esercizio

- a) Progettare un circuito che, ricevuti 3 segnali in ingresso, stabilisca se questi rappresentano, in notazione binaria, un numero divisibile per 2 o per 3. Il circuito restituisce in uscita 1 se l'input rappresenta un numero divisibile per 2 oppure per 3, restituisce 0 in caso contrario.
- b) Progettare un circuito che, ricevuti 4 segnali in ingresso, stabilisca se questi rappresentano, in notazione binaria, un valore divisibile per 2 o per 5. Nei casi in cui i 4 bit non rappresentino alcuna cifra decimale (ossia un valore compreso tra 0 e 9) l'uscita può assumere un valore arbitrario.

1.3 Esercizio

- a) Progettare un circuito combinatorio con quattro ingressi ed un'uscita. L'uscita assume il valore 1 se l'OR dei primi due ingressi è diverso dall'AND dei rimanenti due.
- b) Progettare quindi la versione duale del circuito, ossia un circuito dove l'uscita è data da una porta AND che riceve come ingresso le uscite di un certo numero di porte OR.

Lezione 2

2.1 Esercizio

- a) Progettare un *half-adder*, ossia un circuito combinatorio che somma due bit e genera il bit risultato ed un riporto. L'*half-adder* va realizzato come modulo (sottocircuito) Logisim.
- b) Utilizzando 4 moduli *half-adder*, costruire un circuito che, ricevuto in ingresso un numero binario di 4 cifre, restituisca in uscita il numero binario successivo. Infine trasformare il circuito in un modulo.

2.2 Esercizio

Utilizzare il modulo dell'esercizio precedente, punto b), per costruire un contatore sequenziale a 4 cifre, ossia un circuito sequenziale, senza ingressi e con 4 uscite, che ad ogni ciclo di clock aggiorna il valore in uscita, trasformandolo nel numero binario successivo (dopo aver assunto il valore 15, l'uscita assume 0).

2.3 Esercizio

- a) Costruire un circuito sequenziale con un segnale di ingresso ed uno di uscita che riconosca la stringa 0101: l'uscita assume il valore 1 quando l'ingresso attuale, insieme a quello dei 3 cicli di clock precedenti, forma, nell'ordine temporale, la sequenza 0101; l'uscita assume 0 altrimenti.
- b) Proporre una soluzione alternativa per l'esercizio del punto precedente.

Suggerimento. L'esercizio ammette due soluzioni: una è basata sugli automi a stati finiti e porta ad un circuito con due flip-flop; alternativamente l'esercizio può essere risolto attraverso un circuito che, mediante tre flip-flop, ricorda il segnale d'ingresso dei tre cicli precedenti.

Lezione 3

Docente: dott. Alberto Ciaffaglione

Email: alberto.ciaffaglione@uniud.it

Ricevimento: su appuntamento (tramite email)

URL: <http://users.dimi.uniud.it/~alberto.ciaffaglione/>

3.1 Esercizio

Dati due numeri naturali m ed n , rappresentati in notazione binaria come $\langle m_1 m_0 \rangle$ e $\langle n_1 n_0 \rangle$, effettuare la loro somma binaria (trattare i bit come si farebbe eseguendo l'operazione "in colonna").

3.2 Esercizio

- a) Utilizzando due moduli *half-adder* (esercizio 2.1), realizzare come modulo Logisim un *full-adder*, ossia un circuito combinatorio che somma due bit ed un riporto e genera il bit risultato ed un nuovo riporto.

- b) Dati due numeri naturali m ed n , rappresentati in notazione binaria come $\langle m_1 \ m_0 \rangle$ e $\langle n_1 \ n_0 \rangle$, effettuare la loro somma binaria utilizzando istanze del modulo full-adder progettato al punto precedente.

3.3 Esercizio

Dato l'insieme $A=\{a, b\}$, si possono codificare due suoi sottoinsiemi $X, Y \subseteq A$ mediante le funzioni caratteristiche X_a, X_b, Y_a, Y_b , ove:

$$Z_c \triangleq \begin{cases} 1 & \text{se } c \in Z \\ 0 & \text{se } c \notin Z \end{cases}$$

Ad esempio, $\langle X_a, X_b, Y_a, Y_b \rangle = \langle 1, 0, 1, 1 \rangle$ rappresenta $X=\{a\}$, $Y=\{a, b\}$.

- Realizzare le operazioni di unione, intersezione, differenza e complemento per sottoinsiemi di A mediante moduli Logisim.
- Realizzare la seguente operazione mediante i moduli progettati al punto precedente:

$$F(X, Y) = (X \cap Y) \cup (\bar{X} - Y)$$

Lezione 4

4.1 Esercizio

Utilizzando moduli half-adder e full-adder (esercizi 2.1 e 3.2), costruire un circuito *multiplier*, il quale riceve in ingresso due numeri binari di 3 bit e genera in uscita il prodotto a 6 bit.

4.2 Esercizio

Costruire un dispositivo sequenziale che riceve in ingresso una linea seriale sulla quale vengono trasmessi pacchetti di 3 bit. Il dispositivo genera come uscita un bit che indica se è stato spedito un pacchetto contenente la sequenza 101: l'uscita vale 0 in corrispondenza dei primi due bit di ogni pacchetto; sul terzo bit l'uscita vale 1 se è stata trasmessa la sequenza 101, vale 0 altrimenti.

Lezione 5

5.1 Esercizio

- Costruire un *multiplexer* con un ingresso di controllo, realizzandolo come modulo Logisim.
- Realizzare un multiplexer con due ingressi di controllo, utilizzando tre multiplexer con un ingresso di controllo.

5.2 Esercizio

Utilizzando moduli multiplexer, modificare il circuito contatore sequenziale dell'esercizio 2.2 nel seguente modo: aggiungere un segnale S che imponga l'uscita del contatore al valore 4; più precisamente, quando il segnale S assume il valore 1, il valore di uscita diventi la rappresentazione binaria del numero 4, quando il segnale S assume il valore 0 il contatore riprenda a contare (dal valore 4).

5.3 Esercizio

Costruire un dispositivo sequenziale che riconosca la stringa 101: l'uscita del circuito assume il valore 1 quando l'ingresso attuale, insieme a quello dei 2 cicli di clock precedenti, forma, nell'ordine temporale, la sequenza 101; l'uscita assume il valore 0 altrimenti.

Lezione 6

6.1 Esercizio

- a) Costruire un *decoder* a due ingressi, realizzandolo come modulo Logisim.
- b) Utilizzare moduli del punto precedente e porte AND a due ingressi (oltre a porte NOT) per realizzare un decoder a tre ingressi ed uno a quattro ingressi.

6.2 Esercizio

Modificare il circuito contatore dell'esercizio 5.2 nel seguente modo: dopo aver assunto il valore 15, al ciclo successivo il conteggio riparta dal valore 4.

6.3 Esercizio

Costruire un circuito che simuli il funzionamento di un ascensore a due piani. Il circuito ha due segnali di ingresso che fungono da pulsanti di chiamata al piano e due segnali di uscita che segnalano la presenza dell'ascensore al piano. Si supponga inoltre che l'ascensore impieghi 2 cicli di clock per passare da un piano all'altro.

Lezione 7

7.1 Esercizio

Considerare numeri interi rappresentati in *complemento a due* mediante 4 bit (l'input è cioè fornito per ipotesi in tale notazione).

- a) Realizzare un modulo per il calcolo dell'*opposto* di un numero intero.
- b) Utilizzando il modulo full-adder progettato nell'esercizio 3.2, realizzare un modulo per il calcolo della *somma* di due numeri interi, segnalando l'eventuale underflow/overflow.

- c) Utilizzando i moduli progettati nei due punti precedenti e moduli multiplexer con un segnale di controllo (come in esercizio 5.1), realizzare un'unità *aritmetica* che sappia effettuare la somma e la sottrazione tra due numeri interi e calcolare l'opposto di uno di essi (tali operazioni vengono scelte per mezzo dei segnali di controllo dei multiplexer).

7.2 Esercizio

Costruire un circuito sequenziale con un segnale di ingresso e uno di uscita che riconosca, come ultimi tre simboli letti, la stringa *acb* (restituendo in tal caso il valore 1 e 0 altrimenti).

Lezione 8

8.1 Esercizio

Considerare numeri interi rappresentati in *complemento a due* mediante 4 bit (l'input è cioè fornito per ipotesi in tale notazione).

- a) Realizzare un modulo per verificare la relazione di *minore stretto* ($<$).
- b) Utilizzando il modulo progettato nel punto precedente, un modulo multiplexer con un segnale di controllo e il modulo per il calcolo dell'opposto (esercizio 7.1), realizzare un'unità *aritmetica* che sappia verificare la relazione di minore stretto ($<$) e la relazione di *maggiore stretto* ($<$) (tali verifiche vengono scelte per mezzo del segnale di controllo del multiplexer).

8.2 Esercizio

Risolvere quanto segue mediante registri di 4 flip-flop.

- a) Costruire un circuito che trasformi un segnale parallelo in uno seriale.
Il circuito ha 4 linee di ingresso ed una di uscita: durante il funzionamento le linee di ingresso vengano modificate ogni 4 cicli di clock; tra una modifica dell'ingresso e la successiva, la linea di uscita, in quattro cicli di clock consecutivi, assuma ciascuno dei 4 valori presenti nelle linee di ingresso.
- b) Costruire un circuito che trasformi un segnale seriale in uno parallelo.
Il circuito ha una linea di ingresso e 4 di uscita: le linee di uscita vengano modificate ogni 4 cicli di clock e ad ogni modifica assumano ciascuna uno dei quattro valori presenti in ingresso nei quattro cicli di clock precedenti.

8.3 Esercizio

Costruire un circuito sequenziale con un ingresso ed un'uscita il quale determini se la sequenza d'ingresso negli ultimi 4 cicli di clock sia *palindroma* (cioè se rimane identica quando è letta da destra verso sinistra): l'uscita assuma valore 1 se è stata letta una sequenza palindroma di 4 cifre binarie, 0 altrimenti.