

Esercitazione di laboratorio n. 3

Argomento dell'esercitazione

Progetto di circuiti sequenziali. L'esercitazione è composta di due esercizi, di cui il secondo, in realtà, è solo un'estensione del primo. Gli esercizi vertono sul progetto di un sistema sequenziale (FSM semplice), in grado di riconoscere sequenze di dati su un ingresso.

Esercizio n. 1

Si progetti un circuito sequenziale sincrono dotato, oltre al segnale di clock CLK , di due ingressi di controllo $Reset$ e En , di parallelismo pari ad 1 bit, di un ingresso di dato I e di una uscita U , anch'essi di parallelismo pari ad 1 bit. Il circuito deve comportarsi nel seguente modo:

- Se $Reset = 1$, il circuito si porta nello stato iniziale (con uscita U posta a 0) indipendentemente dal valore degli altri ingressi.
- Se $En = 0$, il circuito rimane nello stato in cui si trova, ignorando il valore dell'ingresso di dato I .
- Se $En = 1$, il circuito campiona il valore presente sul segnale I : l'uscita U deve assumere il valore 1 se gli ultimi 4 valori campionati corrispondono alla sequenza 1011, mentre assume il valore 0 altrimenti. Si osservi che l'ultimo '1' della sequenza può rappresentare anche l'inizio di una nuova sequenza valida.

Si realizzi il circuito come *una macchina di Moore*, che lavori ad una frequenza di clock pari a 50 kHz.

Si implementi, quindi, il circuito sulla scheda FPGA in dotazione, utilizzando i bottoni $BTN3$ e $BTN0$ per i segnali $Reset$ ed En , lo switch $SW0$ per il segnale di dato I ed il led $LD0$ per il segnale di uscita U .

Traccia

Il progetto si articola nelle seguenti fasi:

1. Si realizzi il diagramma degli stati del circuito proposto (su carta).
 2. Si effettui la codifica degli stati, adottando una codifica minima.
 3. Si sintetizzino le funzioni booleane relative alla funzione di transizione degli stati e alla funzione di uscita del circuito.
 4. Tramite il tool Xilinx ISE WebPack, si creino due moduli in un nuovo progetto; essi dovranno essere descritti in codice VHDL, con la definizione di entità e la descrizione delle architetture (comportamentali) relative:
 - a. Alla rete combinatoria sintetizzata.
 - b. Ai flip-flop di tipo D, dotati di ingresso di CLR (sincrono) attivo alto.
- Attenzione:** in fase di creazione del progetto, assicurarsi di selezionare il dispositivo **xc3s200** con package **ft256**.
5. Mediante i moduli appena descritti, si costruisca lo schematico del circuito che implementa il comportamento desiderato.
 6. Si validi, infine, la rete specificata tramite simulazione.

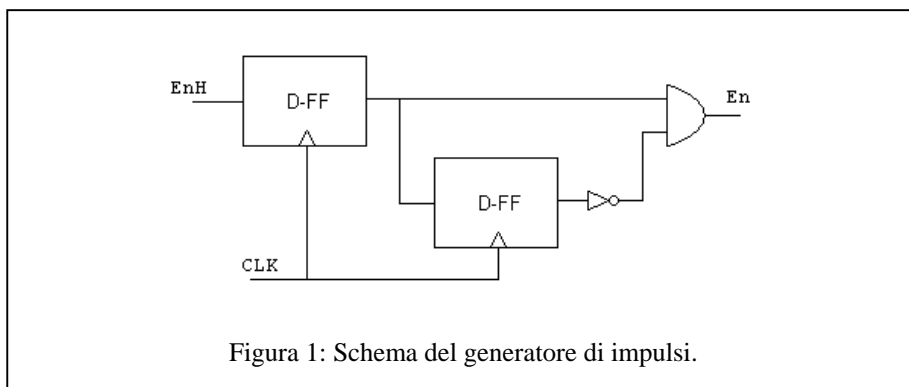
01GSS - Specifica e simulazione dei sistemi digitali

Per l'implementazione su FPGA, poi, occorre specificare due moduli aggiuntivi, che sono:

- Un divisore di frequenza.
- Un generatore di impulsi.

Il divisore di frequenza è necessario per generare il segnale di clock alla frequenza richiesta dall'esercizio (la scheda, infatti, è dotata di un generatore di clock alla frequenza fissa di 50 MHz). Un semplice divisore di frequenza può essere ottenuto tramite un flip-flop di tipo T, che complementa la propria uscita ad ogni colpo (fronte di salita o di discesa) del clock. L'uscita del flip-flop, infatti, è una forma d'onda quadra con frequenza pari alla metà di quella del clock. Il flip-flop T, dunque, è un divisore di frequenza di fattore 2. Ponendo in cascata N flip-flop di tipo T, si ottiene un divisore di frequenza di fattore 2^N . Per generare la frequenza di clock richiesta dall'esercizio (50 kHz) a partire da un segnale di frequenza 50 Mhz, occorrerà quindi una cascata di 10 di questi moduli ($2^{10} \approx 1000$).

Il generatore di impulsi è, in realtà, un riconoscitore di fronti di salita ed è necessario come modulo di interfaccia verso l'operatore umano. Deve essere inserito, infatti, per evitare che ad una singola (in senso umano) pressione del bottone BTN0 che fornisce il segnale di enable EN, vengano campionati più valori sul segnale di dato I. Il generatore di impulsi trasforma il segnale reale proveniente dal bottone in un segnale che assume il valore logico 1 solo nel periodo di clock immediatamente successivo al fronte di salita, facendolo poi ritornare a 0 fino al fronte di salita successivo. A questo scopo si può utilizzare il circuito riportato in Figura 1. Si osservi che il segnale di clock fornito al generatore di impulsi deve essere lo stesso del resto del sistema.



Viene di seguito fornita la corrispondenza tra i nomi dei segnali che intervengono nel sistema e i loro pin sulla FPGA.

I/O Name	Pin
U	K12
CLK	T9
Reset	L14
En	M13
I	F12

Esercizio n. 2

Si modifichi il circuito dell'esercizio precedente in modo tale che i bit di stato siano interpretati come l'espressione in binario puro di un numero intero. Tale numero venga visualizzato sulla cifra più a destra del display a 7 segmenti presente sulla FPGA in dotazione.

Traccia

Rispetto al sistema utilizzato nel primo esercizio, è richiesto il progetto di un modulo aggiuntivo che funzioni da interfaccia verso il display a 7 segmenti.

Per progettare questo modulo, è necessario sapere che:

- Tutti i segnali di controllo verso questo dispositivo sono attivi bassi.
- I segnali che indicano quali segmenti (incluso il punto decimale) illuminare sono condivisi tra tutte le cifre del display (sono chiamati a, b, c, d, e, f, g, e d_p rispettivamente).
- Ogni cifra ha un proprio segnale di abilitazione (indicati con AN3, AN2, AN1, e AN0 rispettivamente).

Per visualizzare un carattere su una cifra del display, dunque, occorre:

- Disabilitare le altre cifre del display, ponendo a 1 i rispettivi segnali di controllo.
- Abilitare i segmenti da illuminare (ponendo i rispettivi segnali a 0) che corrispondono al carattere da visualizzare.
- Abilitare la visualizzazione della cifra.

La sequenza di operazioni appena descritta deve essere ripetuta ciclicamente: ogni cifra del display in pratica viene illuminata solo per una frazione del tempo in cui il display è attivo, ma la sensazione che si avrà è ugualmente quella di una illuminazione costante. Per una buona visualizzazione, occorre che la frequenza di refresh sia compresa tra 60 Hz e 1 kHz.

Il modulo di interfaccia con il display, quindi, deve essere costituito da:

- Un decodificatore: circuito combinatorio che riceve i bit di stato del sistema e genera l'insieme di valori per i segnali che abilitano i segmenti corrispondenti al numero da visualizzare.
- Un generatore di clock alla frequenza di refresh: può essere ottenuto tramite un nuovo divisore, come visto nell'esercizio precedente.
- La circuiteria necessaria per l'abilitazione delle singole cifre.

Si osservi che il circuito che gestisce l'abilitazione delle singole cifre è, in questo caso, molto semplice: poiché è richiesto di visualizzare un numero solo su una cifra, è sufficiente porre al valore costante 1 i segnali che abilitano le altre cifre, e collegare il segnale di clock (alla frequenza di refresh) direttamente con il segnale di abilitazione della cifra che si vuole illuminare. In un caso più complicato (visualizzazione su più cifre), occorrerebbe generare dei segnali di abilitazione delle cifre opportunamente multiplexati, con i segnali di abilitazione dei singoli segmenti sincronizzati con essi.

01GSS - Specifica e simulazione dei sistemi digitali

La tabella che segue fornisce le configurazioni dei segnali di abilitazione dei segmenti a seconda del numero (esadecimale) che si vuole visualizzare.

cifra	a	b	c	d	e	f	g
0	0	0	0	0	0	0	1
1	1	0	0	1	1	1	1
2	0	0	1	0	0	1	0
3	0	0	0	0	1	1	0
4	1	0	0	1	1	0	0
5	0	1	0	0	1	0	0
6	0	1	0	0	0	0	0
7	0	0	0	1	1	1	1
8	0	0	0	0	0	0	0
9	0	0	0	0	1	0	0
A	0	0	0	1	0	0	0
b	1	1	0	0	0	0	0
C	0	1	1	0	0	0	1
d	1	0	0	0	0	1	0
E	0	1	1	0	0	0	0
F	0	1	1	1	0	0	0

Viene di seguito fornita, infine, la corrispondenza tra i nomi dei segnali che intervengono nel sistema e i loro pin sulla FPGA (in aggiunta a quelli dell'esercizio precedente).

I/O Name	Pin
AN3	E13
AN2	F14
AN1	G14
AN0	D14
a	E14
b	G13
c	N15
d	P15
e	R16
f	F13
g	N16
dp	P16