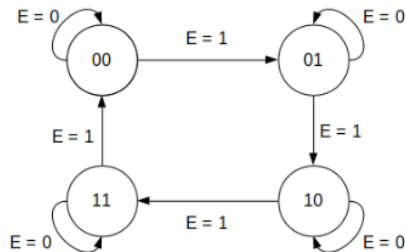


Laboratorio 3 – esercizio 5 – Contatore circolare a 2 bit – soluzione

Vogliamo quindi realizzare un circuito che realizzi il comportamento di un contatore di numeri interi da 0 a 3 generando un output a due bit (q_0q_1) che incrementi il suo valore solo quando il segnale di Enable in input è attivo.

La FSM di tale dispositivo può essere così rappresentata:



La funzione `next_state` del circuito avrà due bit di output e sarà tale che, ad esempio, q_1 che indica la cifra più a destra, quella meno significativa, della rappresentazione binaria del valore del contatore, avrà valore 1 quando il segnale in input è `Enable=0` e lo stato corrente indica un numero dispari (ovvero nel caso di $1=01_2$ e $3=11_2$); avrà inoltre valore 1, quando il segnale `Enable` è attivo (uguale a 1) e lo stato corrente rappresenta un numero pari (ovvero nel caso di 00 e $2=10_2$).

Scrivendo in forma di tabella la funzione `next_state` (coincidente in questo caso con la funzione di output in funzione dello stato corrente e del segnale di `Enable` avremo:

Current State	Next State	
	$E = 0$	$E = 1$
00	00	01
01	01	10
10	10	11
11	11	00

o, in altra forma equivalente,

Enable	$q_0q_1 = 00$	$q_0q_1 = 01$	$q_0q_1 = 10$	$q_0q_1 = 11$
0	$q_0q_1 = 00$	$q_0q_1 = 01$	$q_0q_1 = 10$	$q_0q_1 = 11$
1	$q_0q_1 = 01$	$q_0q_1 = 10$	$q_0q_1 = 11$	$q_0q_1 = 00$

La funzione logica corrispondente a `next_state(q1)` e `next_state(q0)` sarà data dalla somma (OR) delle combinazioni dei tre input booleani (q_0 , q_1 e E) che danno un output a 1:

- del bit più a destra (quello corrispondente a q_1)
 $next_state(q_1) = \bar{E}not(q_0)not(q_1) + not\bar{E}not(q_0)q_1 + E\bar{q}_0not(q_1) + not(E)q_0q_1$
- del bit più a sinistra (quello corrispondente a q_0)
 $next_state(q_0) = \bar{E}not(q_0)q_1 + not(E)q_0not(q_1) + E\bar{q}_0not(q_1) + not(E)q_0q_1$

Per il disegno del circuito, applichiamo dove possibile delle semplificazioni

$$\begin{aligned} q_0 &= \bar{E}(q_0\bar{q}_1 + q_0q_1) + E(\bar{q}_0q_1 + q_0\bar{q}_1) \\ &= q_0\bar{q}_1\bar{E} + q_0q_1\bar{E} + \bar{q}_0q_1E + q_0\bar{q}_1E \\ &= q_0\bar{q}_1 + q_0q_1\bar{E} + \bar{q}_0q_1E \end{aligned}$$

$$\begin{aligned} q_1 &= \bar{E}(\bar{q}_0q_1 + q_0q_1) + E(\bar{q}_0\bar{q}_1 + q_0\bar{q}_1) \\ &= \bar{q}_0q_1\bar{E} + q_0q_1\bar{E} + \bar{q}_0\bar{q}_1E + q_0\bar{q}_1E \\ &= \bar{q}_0(q_1\bar{E} + \bar{q}_1E) + q_0(q_1\bar{E} + \bar{q}_1E) \\ &= q_1\bar{E} + \bar{q}_1E \end{aligned}$$

È disponibile tra le soluzioni la versione .circ ma consigliamo di ridisegnarlo a partire dalle funzioni next_state di q0 e q1 e della sua immagine riportata qui sotto.

P.S. nel disegno del circuito sono stati aggiunti gli oggetti "output" corrispondenti a q0, not(q0), q1 e not(q1). Questo non era richiesto dall'esercizio ma facilita la verifica dei relativi valori durante la simulazione.

Per il disegno del circuito in Logisim, oltre al segnale di Enable da utilizzare come input binario, saranno necessario un clock (disponibile in Logisim tra gli oggetti "Memory"). Il clock sarà uno degli input per ciascuno dei due flip-flop che, preso in input il valore di uscita della porta OR che realizza le funzioni next_state (di q0 o q1, a seconda di quale flip-flop si sta considerando), fornisce come output i nuovi valori di q0 (o q1).

L'oggetto flip-flop di Logisim prevede due uscite distinte per Q e not(Q).

Dopo aver ridisegnato in Logisim il circuito, verificarne il comportamento. Con un click sull'oggetto "clock" si attiva uno step. Con un click sul segnale in input Enable si cambia il suo valore da 0 a 1.

N.B. verificare che tra le opzioni del menu "Simulate" sia abilitata la simulazione!

