

Sistemi di Elaborazione - Soluzioni compito

01.07.2003

2 luglio 2003

Esercizio 1

La soluzione è in figura 1. Come si può notare, come per l'esercizio analogo del compito del primo appello, è necessario uno stato *Start* che è lo stato in cui si trova la macchina all'accensione. Tale stato non è più raggiungibile e dà sempre come uscita 0, qualunque sia l'ingresso presentato. Gli altri quattro stati rappresentano ciascuno un possibile valore precedente degli ingressi. Ricordiamo anche il complemento a due di un numero si ottiene sommando 1 al complemento a 1.

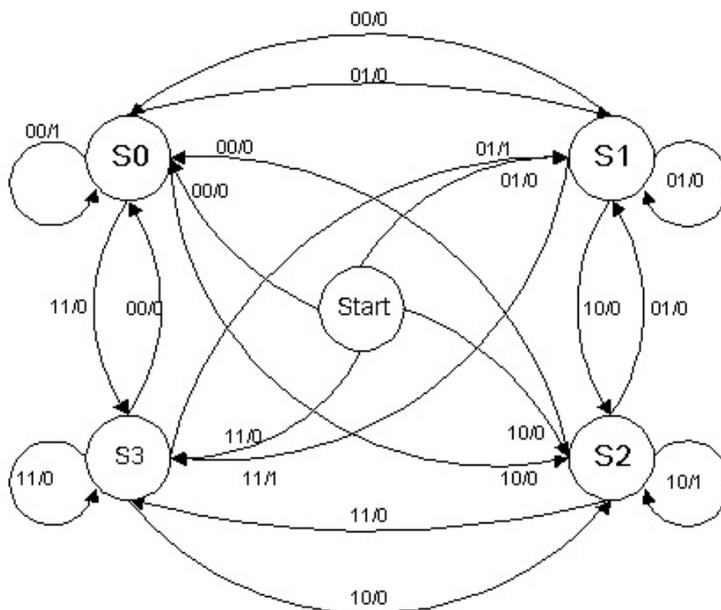


Figura 1: Soluzione esercizio 1

Esercizio 2

Abbiamo a disposizione una PLA and-or, quindi dovremo effettuare una sintesi del tipo SP (somme di prodotti). Ora però abbiamo a disposizione solamente 4 implicant che devono “bastare” per tutte e tre le funzioni. È quindi necessario non effettuare una sintesi minima ma utilizzare il più possibile quegli implicant che sono comuni alle tre funzioni. Quindi:

$$\begin{aligned} F_1 &= A \cdot B + B \cdot \bar{C} + \bar{A} \cdot C \\ F_2 &= \bar{A} \cdot \bar{B} + \bar{A} \cdot C + B \cdot \bar{C} \\ F_3 &= \bar{A} \cdot \bar{B} + A \cdot B + B \cdot \bar{C} \end{aligned}$$

Esercizio 3

Una possibile soluzione è la seguente:

$$\begin{aligned} 0 : & \underbrace{cop \rightarrow K, A \rightarrow A, B \rightarrow B, N \rightarrow C, M \rightarrow D}_{O_1}, 1 \\ 1 : & O_0 (K = 0) \quad 2, (K = 1) \quad 7 \\ 2 : & \underbrace{tsar(A + M) \rightarrow D, 0 \rightarrow A}_{O_2} \quad (C^n = 0) \quad 3, (C^n = 1) \quad 4 \\ 3 : & \underbrace{\bar{C} + 1 \rightarrow C}_{O_3}, 5 \\ 4 : & \underbrace{\bar{D} + 1 \rightarrow D}_{O_4}, 5 \\ 5 : & O_0 (OR(C) = 0) \quad 0, (OR(C) = 1) \quad 6 \\ 6 : & \underbrace{A + D \rightarrow A, C + 1 \rightarrow C}_{O_5}, 5 \\ 7 : & \underbrace{tsar(B + D) \rightarrow B}_{O_6}, 8 \\ 8 : & O_0 (OR(C)C^n = 00), 0, (OR(C)C^n = 10), 9, (OR(C)C^n = 11), 10 \\ 9 : & \underbrace{tsar(B) \rightarrow B, C - 1 \rightarrow C}_{O_7}, 8 \\ 10 : & \underbrace{tdar(B) \rightarrow B, C + 1 \rightarrow C}_{O_8}, 8 \end{aligned}$$

Poiché ci viene richiesto di realizzare la macchina con una parte controllo di tipo *Mealy-ritardato*, ricordandoci che un automa di tipo *Moore* può essere sempre realizzato con un automa di tipo *Mealy-ritardato*, scriviamo il microprogramma TS.

Realizziamo la moltiplicazione per N attraverso somme successive del valore contenuto nel registro D . Prima però (μ -istruzioni 2 e 4) controlliamo il segno di N per ottenere un prodotto col segno corretto.

Per quello che riguarda il caso in cui $cop = 1$, la cosa é più semplice: basta infatti operare N traslazioni aritmetiche (sinistre o destre a seconda del segno di N).

Esercizio 4

Per realizzare effettivamente la parte controllo non ci rimane che effettuare la conversione dell'automa *Moore* dell'esercizio precedente in un'automa *Mealy*, e aggiungere un registro che si interpone fra l'uscita della parte controllo e la parte operativa per passare da *Mealy* a *Mealy-ritardato*.

- 0 : $O_1, 1$
- 1 : $O_0, 2$
- 2 : $(K = 0) O_2, 3, (K = 1) O_6, 7$
- 3 : $(C^n = 0) O_3, 4, (C^n = 1) O_4, 4$
- 4 : $O_0, 6$
- 5 : $(OR(C) = 0) O_1, 2, (OR(C) = 1) O_5, 6$
- 6 : $O_0, 5$
- 7 : $O_0, 8$
- 8 : $(OR(C)C^n = 00) O_1, 1, (OR(C)C^n = 10) O_7, 7, (OR(C)C^n = 11) O_8, 7$

L'inserzione del registro fa' sì che l'automa realizzato a partire da questo microprogramma sia a computazione equivalente rispetto a quello dell'esercizio precedente.

Esercizio 5

Quando si arriva all'etichetta *fine* il registro *ECX* contiene:

$$\sum_{i=1}^n i^2.$$

ovvero la somma dei primi n quadrati, con n il valore contenuto nella locazione 100.

Infatti la subroutine *carre* non fa altro che calcolare il quadrato del contenuto del registro *EAX* (attraverso somme successive) e lasciare il risultato nel registro *EBX*.

Il "programma principale" invece assomma in *ECX* i risultati della procedura *carre*, applicata a tutti i numeri interi minori o uguali a n . Da notare che la subroutine *carre* utilizza anch'essa il registro *ECX*, ma non lo "sporca" perché all'inizio salva il suo contenuto in pila (*PUSHL ECX*) e al termine lo ripristina prelevandolo sempre dalla pila (*POPL ECX*).