Esame di Architetture 1 – canale EO – Andrea Sterbini
appello del 9-6-2004

Obiettivo: si vuol progettare il circuito sequenziale che gestisce un distributore automatico di bibite che ha le seguenti caratteristiche:
- accetta solo monete da **20** e **50** centesimi e accumula **fino ad un massimo di 70** centesimi.
- rifiuta la moneta se con essa il totale accumulato supera i 70 centesimi
- fornisce caffè (che costa 30 centesimi) oppure tè verde (che costa 40 centesimi) quando si preme un tasto di selezione e la cifra raccolta è sufficiente
- non dà il resto (ma ricorda quanto si è pagato in più)

Gli eventi che possono avvenire sono:
- **m**: inserimento della moneta da **20 centesimi**
- **M**: inserimento della moneta da **50 centesimi**
- **C**: pressione del bottone che seleziona il **caffè**
- **T**: pressione del bottone che seleziona il **the verde**

In uscita vengono prodotte le due variabili booleane:
- **E**: (erogazione) che vale 1 se si è premuto un tasto di selezione e può essere erogata la bevanda scelta
- **R**: (rifiutata) che vale 1 se la moneta appena inserita va rifiutata (con essa si superano i 70 cent)

All'inizio il distributore non contiene monete.

Parte 1: (10 punti)

Progettate l'automa che descrive lo stato della macchina e le transizioni di stato al variare degli eventi descritti tenendo conto dell'inserimento delle monete o della pressione sui pulsanti per scegliere le bevande (sono necessari al più 8 stati).

Esempio di una possibile sequenza di eventi con i relativi avvenimenti ed output:

1. **m**: si inserisce una moneta da 20 centesimi	-> non succede niente	(E=0,R=0)
2. **T**: si preme il bottone che seleziona il the	-> non succede niente	(E=0,R=0)
3. **C**: si cambia idea e si preme il pulsante del caffè	-> non succede niente	(E=0,R=0)
4. **M**: si inserisce una moneta da 50 centesimi	-> non succede niente	(E=0,R=0)
5. **m**: si inserisce una moneta da 20 centesimi	-> viene rifiutata	(E=0,R=1)
6. **T**: si preme il bottone che seleziona il the	-> viene erogato il the verde	(E=1,R=0)
7. **m**: si inserisce una moneta da 20 centesimi	-> non succede niente	(E=0,R=0)
8. **C**: si preme il bottone del caffè	-> viene erogato il caffè	(E=1,R=0)

(nota, a questo punto si ha ancora un credito di 20 centesimi)

Parte 2: (5 punti)

Minimizzate l'automa (o dimostrate che è già minimo).

Parte 3: (5 punti)

Codificate come volete gli input e gli stati e calcolate la tabella di transizione del circuito usando flip-flop di tipo **JK** (servono al più 3 FF).

Parte 4: (5 punti)

Realizzate il circuito usando multiplexers 4-a-1 (calcolate solo gli input dei multiplexer).

Parte 5: (10 punti)

Minimizzate le funzioni e realizzate il circuito usando una PLA.

Soluzione

Parte 1: costruzione dell'automa

La macchina sequenziale deve ricordare quanto denaro è stato depositato e quindi sono necessari gli 8 stati corrispondenti a 0, 10, 20, 30, 40, 50, 60, 70 centesimi.
Per ciascuno degli stati si ha una transizione per ciascuno dei 4 possibili eventi:

- m:
  - si passa dallo stato X allo stato X+20 se X<=50, in questo caso la moneta è accettata (R=0) e non viene erogato nulla (E=0)
  - altrimenti si rimane sullo stato X, la moneta viene rifiutata (R=1) e non viene erogato nulla (E=0)
- M:
  - si passa dallo stato X allo stato X+50 se X<=20, in questo caso la moneta è accettata (R=0) e non viene erogato nulla (E=0)
  - altrimenti si rimane sullo stato X, la moneta viene rifiutata (R=1) e non viene erogato nulla (E=0)
- C:
  - si passa dallo stato X allo stato X-30 se X>=30, viene erogata la bevanda (E=1) e R=0
  - altrimenti si rimane nello stato X, non viene erogato nulla (E=0) ed R=0
- T:
  - si passa dallo stato X allo stato X-40 se X>=40, viene erogata la bevanda (E=1) e R=0
  - altrimenti si rimane nello stato X, non viene erogato nulla (E=0) ed R=0

La tabella di transizione è: (in cui con **X/E,R** indico che si va allo stato X mandando in output i valori E,R):

	Evento			
Dallo stato	***m***	***M***	***C***	***T***
'0'	'20'/0,0	'50'/0,0	'0'/0,0	'0'/0,0
'10'	'30'/0,0	'60'/0,0	'10'/0,0	'10'/0,0
'20'	'40'/0,0	'70'/0,0	'20'/0,0	'20'/0,0
'30'	'50'/0,0	'30'/0,1	'0'/1,0	'30'/0,0
'40'	'60'/0,0	'40'/0,1	'10'/1,0	'0'/1,0
'50'	'70'/0,0	'50'/0,1	'20'/1,0	'10'/1,0
'60'	'60'/0,1	'60'/0,1	'30'/1,0	'20'/1,0
'70'	'70'/0,1	'70'/0,1	'40'/1,0	'30'/1,0

Parte 2: minimizzazione dell'automa

Per minimizzare l'automa costruiamo la tabella triangolare e confrontiamo le transizioni per ciascun simbolo in input. Molte coppie (indicate con x) non sono minimizzabili perchè hanno almeno un output diverso.

10	20-30 (x) 50-60 (x)						
20	20-40 (x) 50-70 (x)	30-40 (x) 60-70 (x)					
30	x	x	x				
40	x	x	x	x			
50	x	x	x	x	60-70 (x) 10-20 (x) 0-10 (x)		
60	x	x	x	x	x	x	
70	x	x	x	x	x	x	30-40 (x) 20-30 (x)
	0	**10**	**20**	**30**	**40**	**50**	**60**

Propagando le X ci si accorge che nessuna coppia di stati può essere minimizzata, quindi l'automa è minimo.

Parte 3: codifica di stati e alfabeto e costruzione della tabella di transizione

Per codificare gli stati e i simboli (eventi) in input usiamo ad esempio le codifiche:

Stato	**'0'**	**'10'**	**'20'**	**'30'**	**'40'**	**'50'**	**'60'**	**'70'**
Codice	000	001	010	011	100	101	110	111

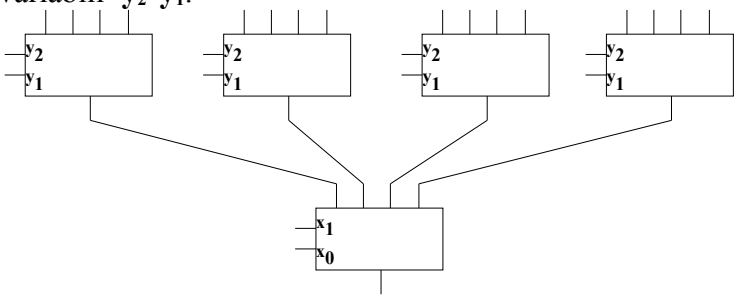
Simbolo	**m**	**M**	**C**	**T**
Codice	00	01	10	11

Quindi la tabella di transizione sarà quella che segue.

Input ***(x_1 x_0)***	***Dallo stato*** ***(y_2 y_1 y_0)***	***Allo Stato*** ***(Y_2 Y_1 Y_0)***	***J_2 , K_2***	***J_1 , K_1***	***J_0 , K_0***	***E***	***R***
00	000	010	0 , -	1 , -	0 , -	0	0
00	001	011	0 , -	1 , -	- , 0	0	0
00	010	100	1 , -	- , 1	0 , -	0	0
00	011	101	1 , -	- , 1	- , 0	0	0
00	100	110	- , 0	1 , -	0 , -	0	0
00	101	111	- , 0	1 , -	- , 0	0	0
00	110	110	- , 0	- , 0	0 , -	0	1
00	111	111	- , 0	- , 0	- , 0	0	1
01	000	101	1 , -	0 , -	1 , -	0	0
01	001	110	1 , -	1 , -	- , 1	0	0
01	010	111	1 , -	- , 0	1 , -	0	0
01	011	011	0 , -	- , 0	- , 0	0	1
01	100	100	- , 0	0 , -	0 , -	0	1
01	101	101	- , 0	0 , -	- , 0	0	1
01	110	110	- , 0	- , 0	0 , -	0	1
01	111	111	- , 0	- , 0	- , 0	0	1
10	000	000	0 , -	0 , -	0 , -	0	0
10	001	001	0 , -	0 , -	- , 0	0	0
10	010	010	0 , -	- , 0	0 , -	0	0
10	011	000	0 , -	- , 1	- , 1	1	0
10	100	001	- , 1	0 , -	1 , -	1	0
10	101	010	- , 1	1 , -	- , 1	1	0
10	110	011	- , 1	- , 0	1 , -	1	0
10	111	100	- , 0	- , 1	- , 1	1	0
11	000	000	0 , -	0 , -	0 , -	0	0
11	001	001	0 , -	0 , -	- , 0	0	0
11	010	010	0 , -	- , 0	1 , -	0	0
11	011	011	0 , -	- , 0	- , 0	0	0
11	100	000	- , 1	0 , -	0 , -	1	0
11	101	001	- , 1	0 , -	- , 0	1	0
11	110	010	- , 1	- , 0	0 , -	1	0
11	111	011	- , 1	- , 0	- , 0	1	0

Parte 4: Costruzione del circuito con multiplexers

Per realizzare le funzioni di eccitazione dei J,K flip-flop e le funzioni di output E,R con multiplexers, visto che abbiamo 5 variabili in input e che un multiplexer 4-a-1 permette di realizzare funzioni di 3 variabili, dobbiamo usare due livelli di 4 e 1 multiplexer.
Scegliamo come ci pare una coppia di variabili da usare come selettori del multiplexer di uscita ed un'altra coppia per i selettori dei 4 flip-flop di ingresso.
Per esempio scegliamo come selettori del multiplexer di output le variabili $\mathbf{x_1}$ $\mathbf{x_0}$ e come selettori dei 4 multiplexer di input le due variabili $\mathbf{y_2}$ $\mathbf{y_1}$.



Costruiamo le tabelle delle funzioni di input del MUX di uscita per le diverse funzioni da realizzare.

LE FUNZIONI DA IMPLEMENTARE SONO: J_2 K_2 J_1 K_1 J_0 K_0 E R

	x_1x_0				x_1x_0				x_1x_0				x_1x_0				x_1x_0				x_1x_0				x_1x_0				x_1x_0			
$y_2y_1y_0$	0 0	0 1	1 0	1 1	0 0	0 1	1 0	1 1	0 0	0 1	1 0	1 1	0 0	0 1	1 0	1 1	0 0	0 1	1 0	1 1	0 0	0 1	1 0	1 1	0 0	0 1	1 0	1 1	0 0	0 1	1 0	1 1
0 0 0	0	0	0	0	0	0	0	0	0	1	0	0	-	-	-	-	1	0	0	0	-	-	-	-	0	1	0	0	-	-	-	-
0 0 1	0	0	0	0	0	0	0	0	0	1	0	0	-	-	-	-	1	1	0	0	-	-	-	-	-	-	-	-	0	1	0	0
0 1 0	0	0	0	0	0	0	0	0	1	1	0	0	-	-	-	-	-	-	-	-	1	0	0	0	0	1	0	1	-	-	-	-
0 1 1	0	0	1	0	0	1	0	0	1	0	0	0	-	-	-	-	-	-	-	-	1	0	1	0	-	-	-	-	0	0	1	0
1 0 0	0	0	1	1	0	1	0	0	-	-	-	-	0	0	1	1	1	0	0	0	-	-	-	-	0	0	1	0	-	-	-	-
1 0 1	0	0	1	1	0	1	0	0	-	-	-	-	0	0	1	1	1	0	1	0	-	-	-	-	-	-	-	-	0	0	1	0
1 1 0	0	0	1	1	1	1	0	0	-	-	-	-	0	0	1	1	-	-	-	-	0	0	0	0	0	0	1	0	-	-	-	-
1 1 1	0	0	1	1	1	1	0	0	-	-	-	-	0	0	0	1	-	-	-	-	0	0	1	0	-	-	-	-	0	0	1	0
MUX	***0***	***0***	***f***	***y_2***	***g***	***f***	***0***	***0***	***y_1***	***h***	***0***	***0***	***0***	***0***	***i***	***1***	***1***	***j***	***k***	***0***	***ny_2***	***0***	***y_0***	***0***	***0***	***ny_2***	***y_2***	***l***	***0***	***m***	***n***	***0***
	E				**R**				**J_2**				**K_2**				**J_1**				**K_1**				**J_0**				**K_0**			

Come si vede dalle tabelle, per alcuni valori di x_1 x_0 la funzione da mandare in input al MUX di uscita (cioè la colonna corrispondente) è una costante oppure è una delle variabili y_2 y_1 y_0 (o un loro complemento).
Negli altri casi (funzioni **f,g,h,i,j,k,l,m,n**) esse si realizzano ciascuna con un MUX descritto dalle tabelle seguenti (le ho appiccicate per occupare meno spazio):

	f				***g***				***h***				***i***				***j***			
	y_2y_1				***y_2y_1***				***y_2y_1***				***y_2y_1***				***y_2y_1***			
y_0	**0 0**	**0 1**	**1 0**	**1 1**	**0 0**	**0 1**	**1 0**	**1 1**	**0 0**	**0 1**	**1 0**	**1 1**	**0 0**	**0 1**	**1 0**	**1 1**	**0 0**	**0 1**	**1 0**	**1 1**
0	0	0	1	1	0	0	0	1	1	1	-	-	-	-	1	1	0	-	0	-
1	0	1	1	1	0	0	0	1	1	0	-	-	-	-	1	0	1	-	0	-
MUX	***0***	***y_0***	***1***	***1***	***0***	***0***	***0***	***1***	***1***	***ny_0***	***-***	***-***	***-***	***-***	***1***	***ny_0***	***y_0***	***-***	***0***	***-***

	k				***l***				***m***				***n***			
	y_2y_1				***y_2y_1***				***y_2y_1***				***y_2y_1***			
y_0	**0 0**	**0 1**	**1 0**	**1 1**	**0 0**	**0 1**	**1 0**	**1 1**	**0 0**	**0 1**	**1 0**	**1 1**	**0 0**	**0 1**	**1 0**	**1 1**
0	0	-	0	-	0	1	0	0	-	-	-	-	-	-	-	-
1	0	-	1	-	-	-	-	-	1	0	0	0	0	1	1	1
MUX	***0***	***-***	***y_0***	***-***	***0***	***1***	***0***	***0***	***1***	***0***	***0***	***0***	***0***	***1***	***1***	***1***

Si noti che **n = not(m)** per cui si può usare un MUX in meno.

Parte 5: Costruzione del circuito con una PLA

Per la parte 5 mostro lo svolgimento solo per una delle funzioni da implementare: J_2
La funzione ha 5 input quindi va minimizzata con una doppia mappa di Karnaugh (oppure col metodo di Quine-McCluskey).
Uso una doppia mappa di Karnaugh suddividendola per i due valori di y_2:

y_2=0				
	x_1x_0			
y_1y_0	**0 0**	**0 1**	**1 1**	**1 0**
0 0	0	1 c	0	0
0 1	0	1 c	0	0
1 1	1 a	0	0	0
1 0	1 a b	1 b	0	0

y_2=1				
	x_1x_0			
y_1y_0	**0 0**	**0 1**	**1 1**	**1 0**
0 0	-	- c	-	-
0 1	-	- c	-	-
1 1	- a	-	-	-
1 0	- a b	- b	-	-

I 3 cubi trovati (indicati dalle tre lettere **a,b,c**) forniscono la minimizzazione:

$$J_2 = y_1\, nx_1\, nx_0 + y_1\, ny_0\, nx_1 + ny_1\, nx_1\, x_0$$

Una volta minimizzate le funzioni $\mathbf{J_2\, K_2\, J_1\, K_1\, J_0\, K_0}$ **E R**, bisogna costruire la PLA.
Vi ricordo che una PLA:

- ha una colonna per ciascuno dei termini contenuti in tutte le (in questo caso 8) funzioni minimizzate
- la colonna che genera un dato termine appare **una sola volta** (anche se il termine appare in più funzioni)
- ha una riga OR per ciascuna funzione in output
- su ogni riga (funzione) di output appare un OR per ciascun termine ad essa appartenente (solo quelli)