

6.3 – Analisi e Sintesi

Il progetto della rete asincrona è dunque molto delicato e richiede l'applicazione di diversi accorgimenti; obiettivo di questo paragrafo è presentare tutta la problematica in modo ordinato, individuando procedimenti di analisi e di sintesi validi per qualsiasi macchina.

Il procedimento di analisi

Il procedimento di analisi di un circuito con retroazioni dirette ha due obiettivi:

- individuare errori di progetto che possono compromettere il buon funzionamento,
- confrontare le specifiche con la descrizione del comportamento dedotta dallo schema logico.

A tal fine è utile eseguire in sequenza i seguenti sei passi.

- **Passo 1 – individuazione delle variabili di stato:** si taglia idealmente sullo schema ogni **retroazione diretta**, denotando con una maiuscola (**variabile di stato futuro**) il segnale a monte del taglio e con la corrispondente minuscola (**variabile di stato presente**) il segnale a valle. Tagliare il percorso di retroazione ed etichettarne le estremità come descritto consente di prescindere dai ritardi introdotti dalla rete combinatoria di aggiornamento dello stato: le variabili di stato futuro costituiscono un'astrazione, rappresentando le uscite di una rete combinatoria ideale priva di ritardi; le variabili di stato presente tengono invece conto dei tempi di propagazione della rete e rappresentano quindi gli unici segnali fisicamente disponibili sul circuito. Se lo schema contiene anche (o solo) dei **latch**, li si elimina idealmente attribuendo una variabile di stato presente all'uscita di ciascuno.

- **Passo 2 – analisi della rete combinatoria d'uscita e della rete combinatoria di aggiornamento dello stato interno:** ispezionando i percorsi di elaborazione presenti nello schema, si scrivono le espressioni delle variabili d'uscita, delle variabili di stato futuro e, se presenti, dei comandi dei latch; in questo ultimo caso si scrivono poi le espressioni dei comandi all'interno dell'equazione caratteristica di ogni latch, ottenendosi così l'espressione della loro variabile di stato futuro.

Al termine, se necessario, si manipolano le espressioni fino ad ottenere forme normali.

- **Passo 3 – individuazione della tabella delle transizioni:** si elencano le configurazioni d'ingresso e di stato sui bordi di una tabella delle transizioni rispettando le condizioni di adiacenza richieste da una mappa di Karnaugh; si tracciano sulla tabella i RR di 1 o di 0 corrispondenti ai termini delle espressioni normali individuate al passo 2 e si completa la tabella con l'altro simbolo logico.

Al termine si controlla se la realizzazione della funzione di aggiornamento presenta o meno il pericolo dell'**alea statica** ("uni" o "zeri" adiacenti non inclusi in uno stesso RR). In caso affermativo si adottano i provvedimenti necessari (v. passo 4 del procedimento di sintesi).

- **Passo 4 – studio delle transizioni e delle condizioni di stabilità:** si evidenziano sulla tabella con un "cerchietto" tutte le situazioni di stabilità della rete, cercando gli incroci in cui la configurazione di stato futuro è uguale a quella di stato presente indicata sulla riga.

Al termine si controlla se il circuito funziona in **modo fondamentale** (per ogni ingresso, da ogni stato stabile deve essere raggiunto uno stato stabile) e se esistono o meno **corse critiche** (transizioni che implicano la variazione contemporanea di più variabili di stato in colonne che presentano diverse condizioni di stabilità). In caso affermativo si adottano i provvedimenti necessari (v. passo 3 del procedimento di sintesi).

- **Passo 5 – individuazione della tabella di flusso:** si sostituisce con una lettera ogni configurazione di stato indicata sulla tabella delle transizioni; una volta ottenuta la tabella di flusso, si eliminano, se presenti, tutte le transizioni multiple (annotando direttamente in ogni incrocio instabile lo stato finale stabile che verrà raggiunto) e tutti gli stati sulla cui riga non compare almeno una condizione di stabilità.

Al termine si controlla se il circuito ha o meno **stati ridondanti** (una riga della tabella di flusso rappresenta uno stato “non necessario” se i suoi incroci sono identici a quelli di un’altra riga) ed in caso affermativo si provvede alla loro eliminazione.

Si controlla anche se il circuito ha o meno **stati irraggiungibili** (una riga della tabella rappresenta uno stato “non utilizzato nel comportamento a regime” se il suo simbolo non compare come stato futuro in nessun altra riga); in caso affermativo si provvede alla loro eliminazione.

➤ **Passo 6 – individuazione del comportamento a regime:** si traccia il grafo degli stati corrispondente alla tabella di flusso, si deduce una stringata descrizione a parole del comportamento e la si confronta con le iniziali specifiche del circuito.

CASO DI STUDIO – La PLA di figura ha chiaramente due retroazioni dirette: il circuito è quindi di tipo asincrono.

Passo 1 - Denotiamo con x_1, x_2 le variabili d’ingresso, con z la variabile d’uscita, con y_1, y_2 le variabili di stato presente e con Y_1, Y_2 quelle di stato futuro

Passo 2 - Mettendo in conto la programmazione evidenziata sulle due matrici, si ottiene la seguente descrizione algebrica del comportamento:

- $Y_1 = x_2'y_2 + x_1y_2 + x_1'x_2y_1$
- $Y_2 = x_2'y_2 + x_1y_2 + x_1'x_2y_1'$
- $z = y_1$

Le espressioni, già di tipo SP, non richiedono alcuna manipolazione.

Passo 3 - Lo studio della funzione di aggiornamento dello stato interno sulle mappe di Karnaugh mostra che nella progettazione del circuito si è fatto riferimento al criterio del costo minimo: tutti i RR rettangolari hanno infatti dimensioni massime ed è stato inoltre impiegato il n° minimo di RR.

Non si è però tenuto conto del pericolo dell’alea statica: nella copertura di Y_1 manca infatti il RR corrispondente al termine y_1y_2 , nella copertura di Y_2 il RR corrispondente al termine $y_1'y_2$.

La disponibilità di due AND all’interno della PLA consente di provvedere alla eliminazione di questa potenziale causa di malfunzionamento.

Passo 4 - Preso atto che la funzione d’uscita vale 1 solo in corrispondenza delle ultime due righe, è possibile riportare il tutto sulla tabella delle transizioni ed evidenziare poi le situazioni di stabilità.

Ogni colonna contiene due condizioni di stabilità e due transizioni che le raggiungono: il circuito funziona dunque in modo fondamentale.

In ogni transizione è inoltre richiesta la modifica di un solo bit di stato: non ci sono dunque corse critiche.

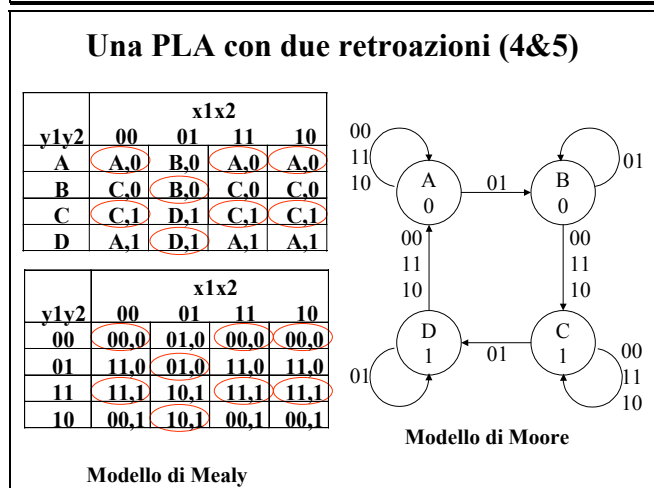
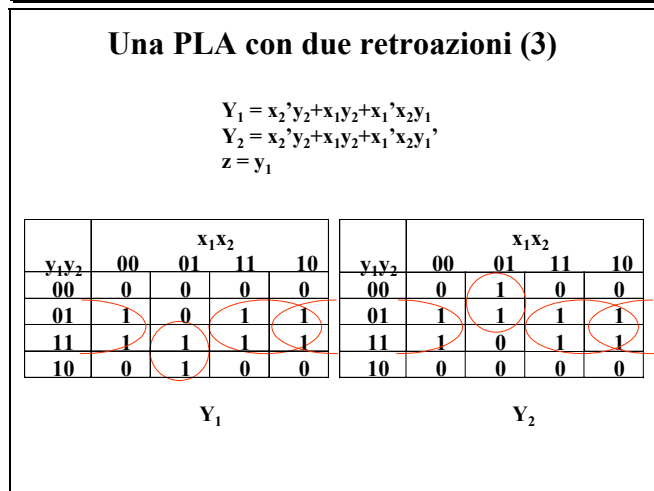
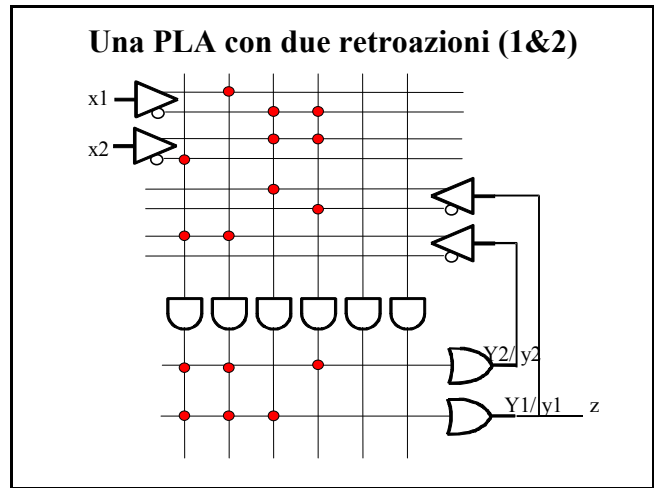
A questo punto conviene proseguire l’analisi sulla tabella di flusso, denotando con A la configurazione di stato 00, con B la configurazione 01, con C la configurazione 11 e con D la configurazione 10.

Passo 5 – Controllando le colonne, si può agevolmente concludere che non ci sono transizioni multiple.

Controllando le righe si può prendere atto che non ci sono stati ridondanti e irraggiungibili.

L’unica proprietà notevole della tabella è che l’uscita dipende dallo stato e non dagli ingressi.

Passo 6 – Il grafo degli stati può dunque essere tracciato seguendo il modello di Moore.



Da quest'ultimo si deduce che il simbolo d'ingresso 01 non modifica mai il simbolo d'uscita, cosa che invece capita con il simbolo successivo, qualsiasi esso sia.

In realtà bisognerà escludere 10 dagli ingressi possibili dopo 01. Il comportamento del circuito è dunque quello di produrre un cambiamento d'uscita in risposta alle sequenze d'ingresso 01-00 e 01-11.

CASO DI STUDIO – Nel circuito a NAND della figura a lato sono presenti due retroazioni dirette: ci troviamo dunque di fronte ad una rete asincrona con quattro stati.

In figura sono indicate sia le espressioni a NAND dei segnali d'uscita e di stato futuro, sia le loro corrispondenti espressioni SP. E' facile verificare che le espressioni sono minime e che non esiste il pericolo dell'alea statica.

Immediatamente sopra alle espressioni è riportata la tabella delle transizioni.

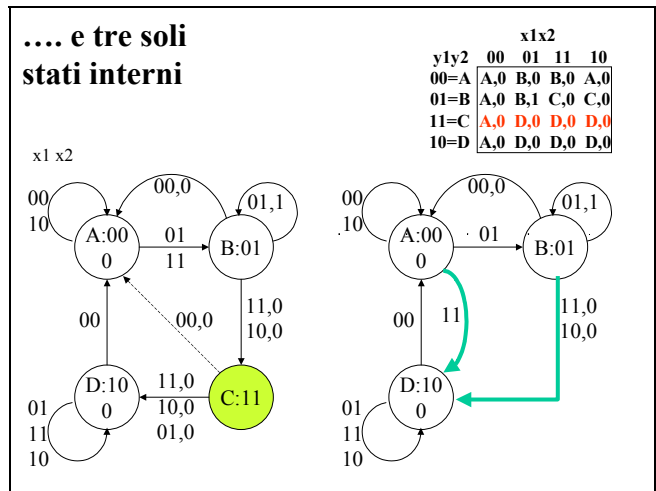
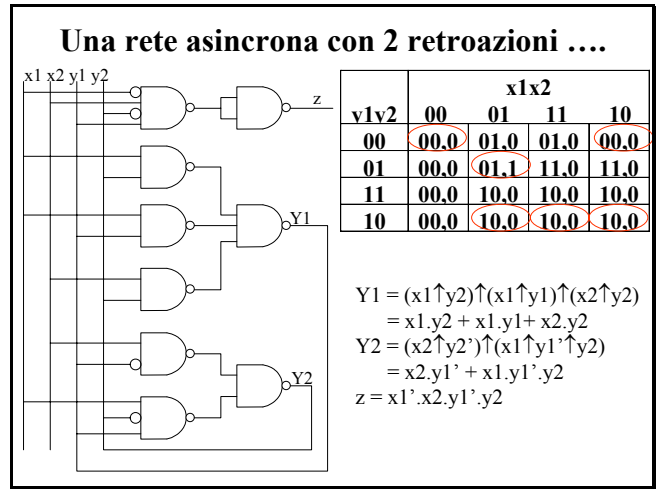
Una volta evidenziate le situazioni di stabilità, si vede subito che la riga 11 ne è priva. Lo stato 11 è stato dunque introdotto ad arte solo per compiere in modo ordinato (cioè cambiando un solo bit alla volta) la transizione da 00 a 10 presente in colonna 11 e le transizioni da 01 a 10 richieste nelle colonne 11 e 10.

La corrispondente tabella di flusso ed il grafo relativo sono evidenziati in figura, unitamente al grafo che si ottiene eliminando lo stato di transizione C.

L'automa minimo che modella il circuito ha dunque solo tre stati.

Si noti che in questo caso è indispensabile porre un vincolo sulle sequenze d'ingresso: se infatti si potesse verificare la configurazione 10 in presenza dello stato B, il nuovo stato stabile dipenderebbe dall'ordine con cui i segnali x1 e x2 hanno modificato il loro valore (D se x1 si modifica prima di x2, A se invece è x2 il primo a cambiare di valore).

Si noti infine che l'eliminazione degli stati ridondanti può non comportare una riduzione del numero di retroazioni: per la legge della codifica, quest'ultimo deve infatti essere un intero maggiore o uguale al logaritmo in base 2 del primo. Un importante accorgimento di progetto è che gli stati "in più" siano resi instabili per ogni ingresso: operando in questa maniera si può infatti evitare che una inizializzazione aleatoria del circuito (cosa che capita ad esempio quando si fornisce alimentazione) ne blocchi temporaneamente il funzionamento.



CASO DI STUDIO – Dall'analisi di uno schema contenente gate AND, OR, NOT e caratterizzato da due ingressi, da due uscite e da due retroazioni dirette, sono state individuate le seguenti espressioni:

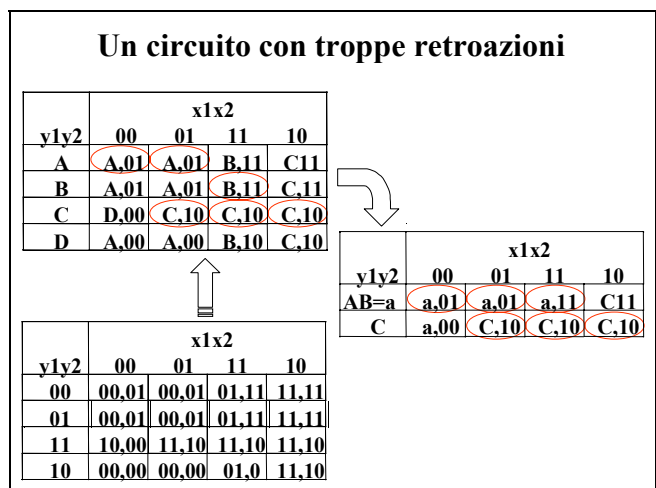
- $Y1 = y1.y2 + x1.x2'$ $Y2 = x1 + x2.y1.y2$
- $z1 = x1 + x2.y1.y2$ $z2 = y1'$

La tabella delle transizioni e la corrispondente tabella di flusso sono indicate in figura.

Una volta evidenziate sulla tabella di flusso le situazioni di stabilità, è possibile trarre due conclusioni.

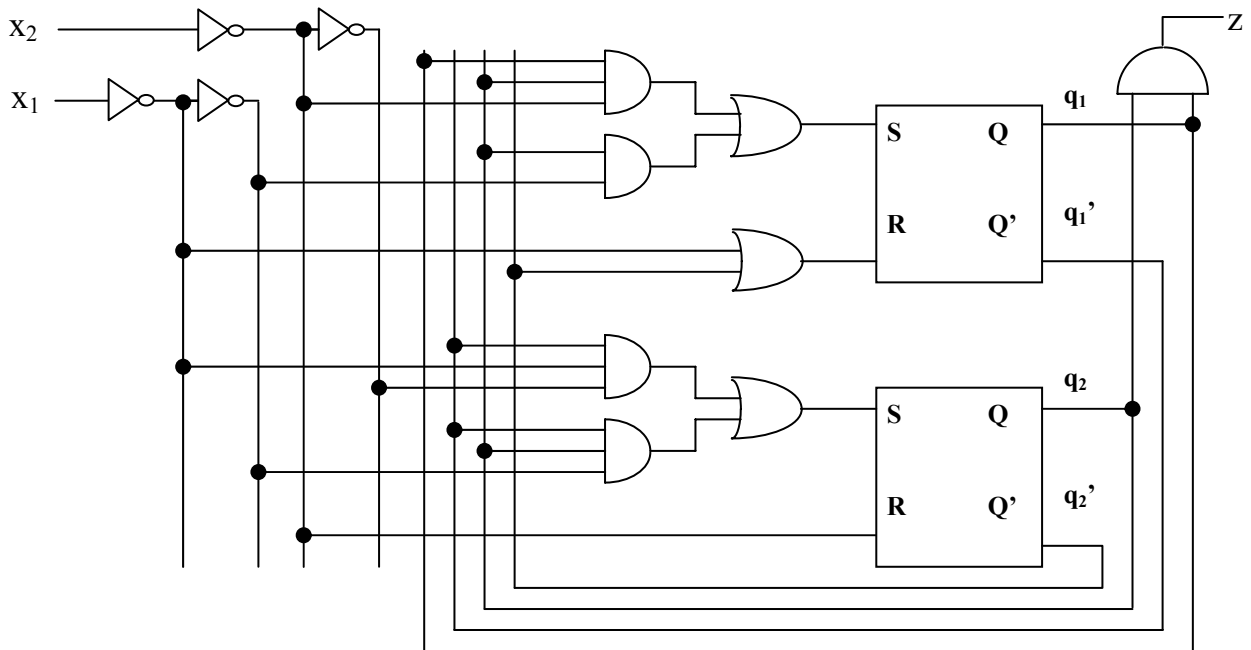
Lo stato D, mai stabile, è utilizzato solo nella transizione multipla che, a fronte dell'ingresso 00, porta la rete dallo stato C allo stato A: ponendo direttamente il simbolo A nella prima colonna della riga C, lo stato D diventa irraggiungibile e deve quindi essere eliminato.

Esistono inoltre due righe identiche: la A e la B. Gli stati A e B sono dunque equivalenti ed è di conseguenza possibile costruire una **tabella minima**, in cui la coppia di righe A,B è sostituita dalla sola riga "a"; tale simbolo deve ovviamente essere impiegato al posto di A e di B nelle indicazioni di stato futuro.



Esercitazione N. 12

Si esegua l'analisi della rete sequenziale asincrona mostrata in figura:



DOMANDA N. 1 - Si scrivano le espressioni SP delle funzioni che comandano gli ingressi dei latch S-R e dell'uscita.

$S_1 =$

$R_1 =$

$S_2 =$

$R_2 =$

$z =$

DOMANDA N. 2 - A partire dalle espressioni SP ottenute al punto precedente si ricavano le corrispondenti mappe di Karnaugh indicando chiaramente i raggruppamenti rettangolari.

		X_1X_2			
$q_1 q_2$		00	01	11	10
00					
01					
11					
10					

S_1

		X_1X_2			
$q_1 q_2$		00	01	11	10
00					
01					
11					
10					

S_2

		X_1X_2			
$q_1 q_2$		00	01	11	10
00					
01					
11					
10					

R_1

		X_1X_2			
$q_1 q_2$		00	01	11	10
00					
01					
11					
10					

R_2

DOMANDA N. 3 – Utilizzando l'equazione caratteristica in forma SP del latch e l'espressione dell'uscita z si ricavi la tabella delle transizioni

$q_1 q_2$	$x_1 x_2$				z
	00	01	11	10	
00					
01					
11					
10					

$Q_1 Q_2$

DOMANDA N. 4 – Si tracci il grafo degli stati escludendo eventuali stati e transizioni non utili all'analisi del comportamento

DOMANDA N. 5 – Qual è la più breve sequenza di ingressi che garantisce l'attivazione dell'uscita indipendentemente dallo stato di partenza?

DOMANDA N. 6 – Osservando le mappe tracciate alla domanda 2, si individui la presenza di configurazioni inappropriate per un latch S-R. Quali sono queste configurazioni e per quali valori di stato presente ed ingressi ($q_1 q_2 x_1 x_2$) si presentano?

DOMANDA N. 7 – Osservando la tabella ed il grafo, dimostrare che la presenza delle configurazioni inappropriate individuate alla domanda precedente non influisce sul comportamento della rete.

Il procedimento di sintesi

Il procedimento di sintesi di una rete sequenziale asincrona è articolato in cinque passi e consente di individuare uno schema logico che presenta un dato comportamento.

- **Passo 1 – individuazione del grafo degli stati**
- **Passo 2 – definizione della tabella di flusso:**
- **Passo 3 – codifica degli stati e definizione della tabella delle transizioni**
- **Passo 4 – sintesi della parte combinatoria**
- **Passo 5 – disegno dello schema logico**

Per fissare le idee sulle attività progettuali richieste in ciascun passo conviene fare riferimento ad un caso. Consideriamo dunque di nuovo il progetto di una lampada da tavolo, parzialmente affrontato in precedenza (v. pag. 46).

La prima cosa da mettere in evidenza è la necessità di una chiara descrizione del comportamento della macchina da progettare.

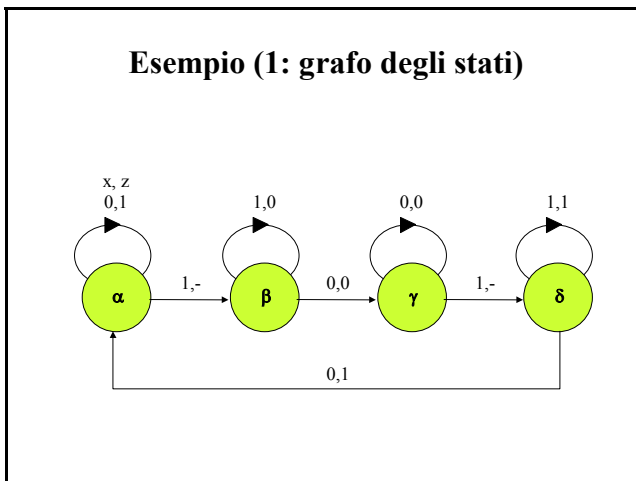
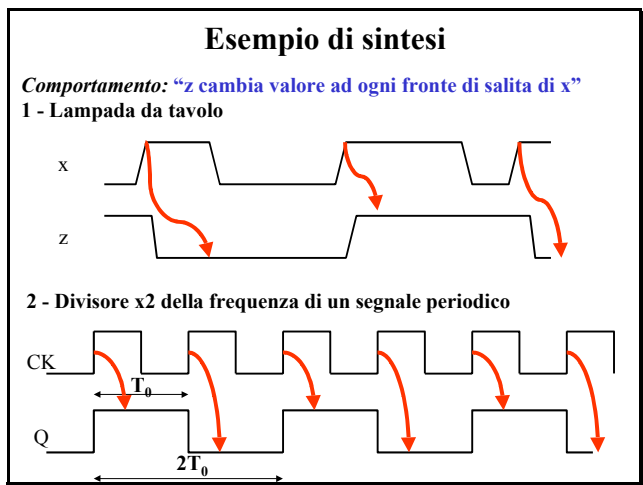
Di notevole aiuto possono essere forme d'onda che illustrino situazioni significative della relazione ingresso/uscita.

In figura è evidenziato che il **divisore per 2** della frequenza di un segnale periodico, un circuito di larghissimo impiego all'interno di un calcolatore, ha lo stesso comportamento della lampada.

Dalla descrizione del comportamento bisogna saper dedurre dapprima il grafo degli stati e poi la tabella di flusso.

Due commenti di carattere del tutto generale a quanto abbiamo già fatto nei primi due passi di progetto di questa macchina asincrona.

- Nel tracciare il grafo bisogna ricordarsi sempre di imporre la stabilità di ogni stato per la configurazione d'ingresso con cui lo si raggiunge (non devono esserci transizioni multiple).
- Nell'indicare sui rami la configurazione d'uscita occorre tener presente che il transitorio è molto breve: quando un segnale d'uscita ha un valore in corrispondenza di una certa situazione di stabilità ed il valore complementare nella situazione che si dovrà raggiungere, la scelta migliore è quella di porre una condizione d'indifferenza sul ramo che li congiunge (sulle mappe delle variabili d'uscita si deciderà, al passo 4, se è più conveniente usare il vecchio o il nuovo valore).



Esempio (2: tabella di flusso)

		x	
		0	1
stato	α	α,1	β,-
	β	γ,0	β,0
	γ	γ,0	δ,-
	δ	α,1	δ,1

Anche i passi 3 e 4 richiedono accorgimenti di carattere del tutto generale.

- Per eliminare a priori il pericolo delle corse critiche, occorre imporre l'adiacenza delle configurazioni che codificano ogni stato presente ed il suo stato futuro in tutte le colonne che contengono più situazione di stabilità; il procedimento da seguire è già stato illustrato a pag. 103.
- Per eliminare a priori il pericolo dell'alea statica, la copertura delle funzioni di stato futuro deve garantire che ogni coppia di 1 adiacenti (se si cerca una espressione SP) o di 0 adiacenti (se si cerca una espressione PS) sia racchiusa in almeno un RR; nei casi in cui occorra aggiungere un RR di questo tipo alla copertura minima, è senz'altro utile attribuirgli dimensioni massime.

Esempio (3: tabella delle transizioni)

		y_2	
		0	1
y_1	0	$\alpha \rightarrow \beta$	
	1	$\delta \leftarrow \gamma$	

Grafo delle adiacenze e mappa di codifica

		x	
		0	1
y_1, y_2	$\alpha: 00$	00,1	01,-
	$\beta: 01$	11,0	01,0
	$\gamma: 11$	11,0	10,-
	$\delta: 10$	00,1	10,1

Y_1, Y_2, z

Esempio (4: espressioni)

Ipotesi: si desiderano reti minime di tipo SP

		$y_1 y_2$			
		00	01	11	10
x	0	1	1	0	
	1	0	1	1	

		$y_1 y_2$			
		00	01	11	10
x	0	1	1	0	
	1	1	1	0	0

		$y_1 y_2$			
		00	01	11	10
x	0	1	0	0	1
	1	-	0	-	1

$Y_1 = x' \cdot y_2 + x \cdot y_1 + y_2 \cdot y_1$
 $Y_2 = x' \cdot y_2 + x \cdot y_1' + y_2 \cdot y_1'$
 $z = y_2'$

Una volta trovate le espressioni, l'individuazione dello schema che le realizza ha due aspetti che occorre prendere in considerazione anche nel progetto di ogni altra macchina asincrona.

- La realizzazione di termini uguali deve essere attribuita ad un unico gate, sfruttando la sua proprietà di avere fan-out maggiore di 1.
- La necessità di dover elaborare, in generale, sia le variabili di stato, sia i loro complementi richiede l'introduzione di NOT e di ulteriori retroazioni; una alternativa interessante è quella di impiegare latch SR sui rami in retroazione, manipolando le espressioni di stato futuro per ricavare le *funzioni di eccitazione*, cioè realizzare le reti combinatorie che pilotando gli ingressi S ed R inducano le transizioni di stato desiderate. Questo passo aggiuntivo è ripagato dal vantaggio immediato di disporre di segnali complementari molto ben allineati, ed inoltre non richiede (a differenza della sintesi diretta delle funzioni di stato futuro) di eliminare le alee statiche. Ricordiamo infatti che l'alea è una "mancanza" temporanea di un valore logico che dovrebbe conservarsi costante. Un latch SR con comandi attivi sul livello logico "1" non soffre se pilotato con reti che presentano alee di 1, poiché l'assenza temporanea di un comando di "set" o di "reset" viene semplicemente interpretata come comando di mantenere quanto è già memorizzato.

Esempio (5: schema con retroazioni dirette)

$Y_1 = x' \cdot y_2 + x \cdot y_1 + y_2 \cdot y_1$
 $Y_2 = x' \cdot y_2 + x \cdot y_1' + y_2 \cdot y_1'$
 $z = y_2'$

Esempio (5: schema con latch SR)

$Y_1 = x' \cdot y_2 + x \cdot y_1 + y_2 \cdot y_1$
 $= (x' \cdot y_2) + (x + y_2) \cdot y_1$
 $= (x' \cdot y_2) + (x' \cdot y_2)' \cdot y_1$
 $= S_1 + R_1' \cdot y_1$

$Y_2 = x' \cdot y_2 + x \cdot y_1' + y_2 \cdot y_1'$
 $= x \cdot y_1' + (x' + y_1') \cdot y_2$
 $= x \cdot y_1' + (x \cdot y_1) \cdot y_2$
 $= S_2 + R_2' \cdot y_2$

$z = y_2'$

Grafi degli stati primitivi e non primitivi

Come è stato detto più volte, all’arrivo di ogni nuovo simbolo d’ingresso la rete asincrona deve passare da una precedente situazione di stabilità ad una nuova situazione di stabilità: ciò può avvenire con o senza modifica dello stato e dell’uscita.

Nell’esecuzione del primo passo del procedimento di sintesi, per rispettare con maggiore facilità il comportamento richiesto, è spesso utile prevedere che ogni modifica d’ingresso produca una modifica di stato.

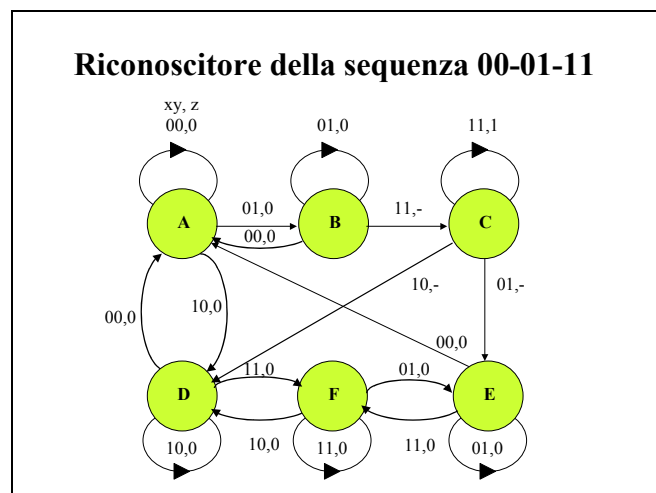
Il grafo risultante è detto **primitivo** ed ha la proprietà che ogni suo nodo è stabile per **una sola** configurazione d’ingresso.

ESEMPIO – Si deve progettare il grafo di una rete sequenziale asincrona a due ingressi che deve riconoscere il verificarsi della sequenza d’ingresso 00-01-11 portando a 1 la sua uscita z; ogni altra sequenza deve determinare z = 0. Per ipotesi gli ingressi cambiano di valore uno solo alla volta.

In figura è mostrata l’utilità del grafo primitivo: è infatti possibile tracciare e collegare di colpo i tre stati e le tre condizioni di stabilità in cui la macchina dovrà passare se si verifica la sequenza cercata.

E’ anche facile completarlo. Se la macchina è in B e viene 00, la sequenza è errata, ma lo 00 può essere l’inizio di una sequenza buona: occorre dunque riportarsi in A. Se la macchina è in C (ed ha quindi appena identificato una sequenza buona) e si verifica 01, bisogna cambiare stato, ma è sbagliato andare in B: un successivo ingresso 11 riporterebbe in C con uscita 1 per la sequenza errata 11-01-11. Occorre aggiungere dunque un nuovo stato D, stabile per 10 e con uscita 0: per 00 da D occorre riportarsi in A.

L’ingresso 01 in C caratterizza una sequenza sbagliata e determina la transizione C-E con uscita 0; al solito per 00 da E bisogna riportarsi in A. Lo stato F, stabile per 11 con uscita 0, una volta connesso agli stati D, E, completa l’identificazione delle sequenze errate.



In generale un grafo primitivo ha un numero di stati superiore a quello strettamente necessario. Se il comportamento lo consente, rimanere in uno stesso stato per più ingressi (e, se richiesto, con uscite diverse da ingresso a ingresso) può infatti portare ad una **riduzione** del loro numero. Il vantaggio è un grafo più “leggibile”. In certi casi si ottiene anche una realizzazione più “economica” (può infatti capitare che l’intero immediatamente superiore al logaritmo del numero di stati del grafo **ridotto** sia più piccolo di quello del grafo primitivo e consenta quindi di impiegare un più basso numero di retroazioni).

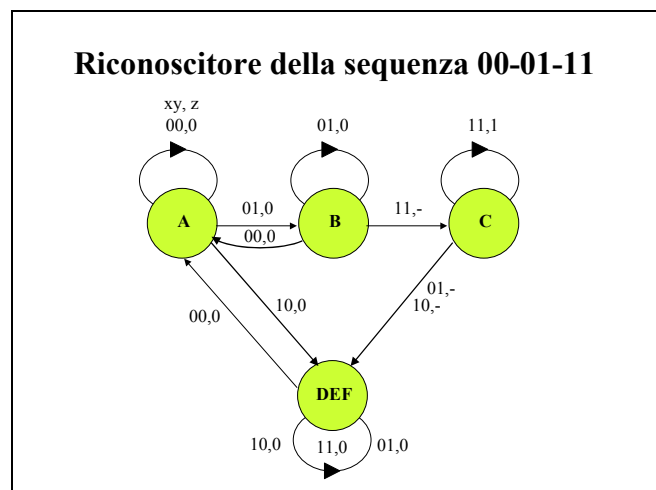
Le situazioni in cui uno stato può essere reso stabile per più ingressi sono spesso agevolmente identificabili durante il tracciamento del grafo degli stati.

ESEMPIO – Nel riconoscitore precedentemente esaminato una sola sequenza è “buona” e tutte le altre sono “cattive”.

Una volta usciti dallo stato C, o con ingresso 01 o con ingresso 10, si può andare direttamente in un unico stato di “scarto” (in figura denominato DEF per ricordare che congloba i “riassunti” prima separatamente assegnati agli stati D, E, F) e rimanervi poi fino a quando non si verifica l’ingresso 00, cioè il primo simbolo della sequenza buona.

L’unico stato futuro di DEF è dunque A, in cui occorre portarsi non appena si presenta il primo simbolo della sequenza “buona”.

L’ingresso 10 in A individua una sequenza “cattiva” e deve quindi riportare la macchina nello stato DEF.



La riduzione del numero di stati può però essere fatta anche, in modo sistematico, sulla tabella di flusso. Una regola per individuare stati “equivalenti” durante il procedimento di analisi è già stata enunciata a pag. 115.

In sede di sintesi occorre allargarla, per tener conto della possibile presenza di condizioni d’indifferenza sullo stato futuro e sulle uscite:

- **“due o più righe di una tabella di flusso possono essere compresse in un’unica riga se, per ogni ingresso, presentano simboli di stato futuro e di uscita identici a meno di condizioni d’indifferenza”.**

ESEMPIO – La tabella di flusso del riconoscitore a 6 stati è la più a sinistra di quelle indicate in figura.

Si noti che le righe DEF soddisfano la regola precedente. Esiste dunque una tabella di flusso (quella al centro in figura) che esprime lo stesso comportamento con soli 4 stati. Per compattezza di simbologia, alla classe di stati {D, E, F} è stato assegnato il nome S.

Si noti ora che anche gli stati della classe {A, B} soddisfano la regola. Esiste dunque una terza versione della tabella (quella più a destra in figura) in cui sono impiegati solo tre stati: il nome I sostituisce i precedenti nomi A e B.

La versione a 6 stati richiede 3 bit di codifica e quindi 3 retroazioni; naturalmente prima di tracciare la tabella delle transizioni occorre aver ben studiato il grafo delle adiacenze.

Le versioni a 4 ed a 3 stati richiedono solo 2 bit di codifica. Si noti che nell’incrocio C,00 dell’automata minimo è possibile inserire la transizione verso I al posto della condizione d’indifferenza: esistono così due colonne (ingresso 00 e ingresso 10) caratterizzate da una sola condizione di stabilità. Le uniche adiacenze richieste nella codifica sono dunque I-C e C-S: può ad esempio andar bene porre I = 00, C = 01 e S = 11.

Il circuito avrà anche lo stato 10 ed è opportuno che non sia mai stabile. Sulla tabella delle transizioni di figura sono state previste una transizione verso 00 nella prima colonna e transizioni verso 11 nelle altre tre.

Tabelle di flusso

Automata minimo

	00	01	11	10		00	01	11	10		00	01	11	10
A	A,0	B,0	-,-	D,0	A	A,0	B,0	-,-	S,0	I	I,0	I,0	C,-	S,0
B	A,0	B,0	C,-	-,-	B	A,0	B,0	C,-	-,-	C	-,-	S,-	C,1	S,-
C	-,-	F,-	C,1	D,-	C	-,-	S,-	C,1	S,-	S	I,0	S,0	S,0	S,0
D	A,0	-,-	E,0	D,0	S	A,0	S,0	S,0	S,0					
E	-,-	F,0	E,0	D,0										
F	A,0	F,0	E,0	-,-										

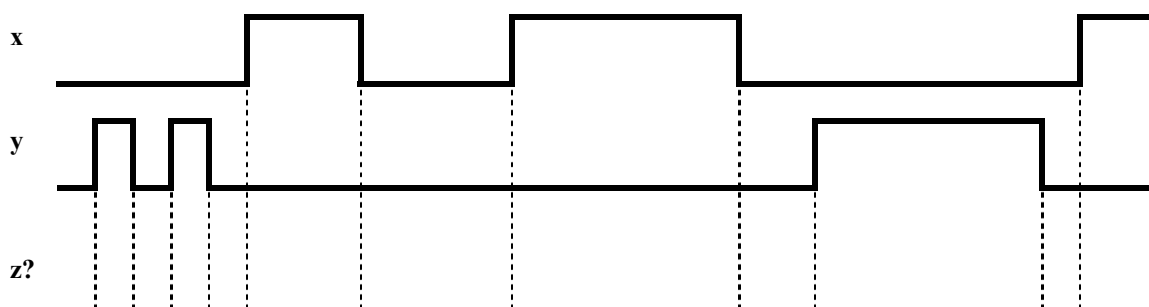
I – C		00	01	11	10
	I:00	00	00	01	11
S	C:01	00	11	01	11
	S:11	00	11	11	11
	10	00	11	11	11

Esercitazione N. 13

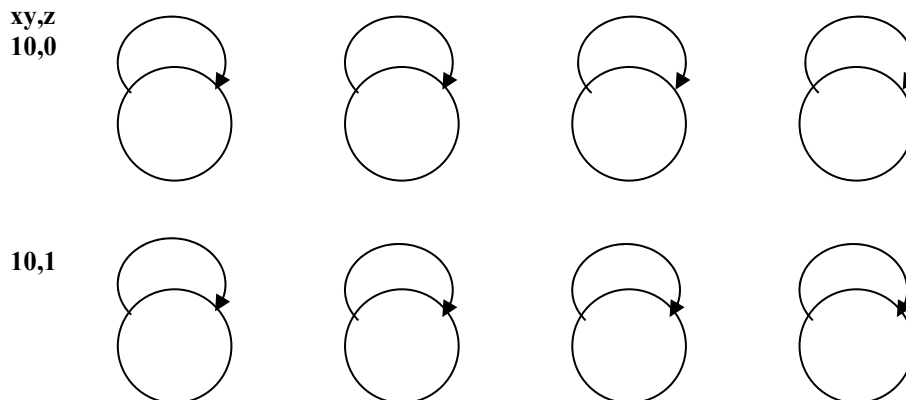
Una rete sequenziale asincrona ha due ingressi x,y ed una uscita z. Gli ingressi non cambiano mai di valore contemporaneamente e non presentano mai entrambi il valore 1.

L’uscita può cambiare di valore solo in corrispondenza dei fronti di salita di x. Il valore che z deve assumere e poi mantenere costante fino al fronte successivo è 0 se nel precedente intervallo x=0 y non ha modificato il suo valore, 1 nel caso opposto.

DOMANDA N.1 - Individuare la forma d’onda del segnale d’uscita z in corrispondenza delle forme d’onda dei segnali di ingresso x,y indicate in figura.



DOMANDA N. 2- Completare il grafo primitivo degli stati (ogni stato deve essere stabile per una sola configurazione d'ingresso)



DOMANDA N. 3 – Tracciare la tabella di flusso

	xy			
s	00	01	11	10
A				
B				
C				
D				
E				
F				
G				
H				

s*, z

DOMANDA N. 4 – Individuare una codifica appropriata degli stati interni

	bc			
a	00	01	11	10
0				
1				

DOMANDA N. 5 – Individuare le espressioni SP delle variabili di stato futuro, eliminando a priori alee statiche nella realizzazione.

	xy			
bc	00	01	11	10
00				
01				
11				
10				

a=0

	xy			
bc	00	01	11	10
00				
01				
11				
10				

a=1

A = G1(a,b,c,x,y) =

	xy			
bc	00	01	11	10
00				
01				
11				
10				

a=0

	xy			
bc	00	01	11	10
00				
01				
11				
10				

a=1

B = G2(a,b,c,x,y) =

	xy			
bc	00	01	11	10
00				
01				
11				
10				

a=0

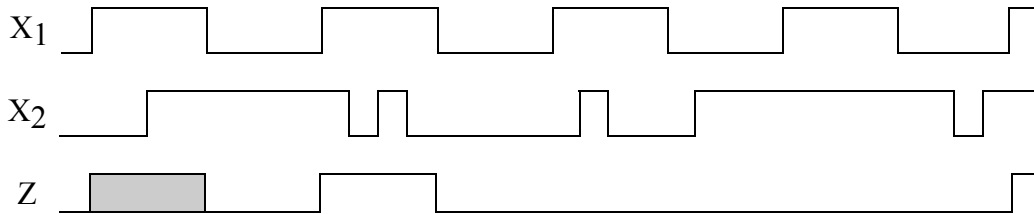
	xy			
bc	00	01	11	10
00				
01				
11				
10				

a=1

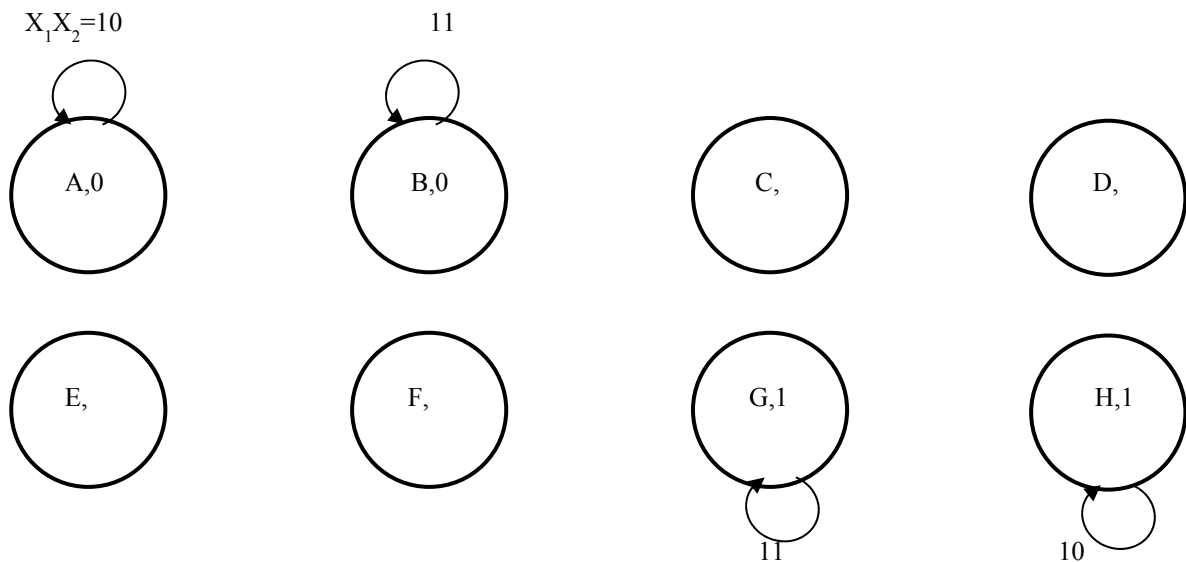
C = G3(a,b,c,x,y) =

Esercitazione N. 14

Una rete sequenziale asincrona è caratterizzata da due segnali di ingresso X_1, X_2 (i quali non cambiano mai contemporaneamente) e da un segnale di uscita Z . Quando il segnale X_1 è disattivo (livello logico 0), Z deve assumere il valore 0. Quando il segnale X_1 è attivo (livello logico 1), Z deve assumere l'ultimo valore presentato dal segnale X_2 nel precedente intervallo di attivazione di X_1 .



DOMANDA N. 1 – Si completi il grafo primitivo degli stati in accordo al modello di Moore.



DOMANDA N. 2 – Si determini una codifica degli stati atta a garantire l'adiacenza delle configurazioni associate ad ogni coppia stato presente – stato futuro e si tracci la tabella delle transizioni secondo il modello di Mealy.

Mappa di codifica

y_1	y_2y_3			
	00	01	11	10
0				
1				

Tabella delle transizioni

stato	$y_1y_2y_3$	X_1X_2			
		00	01	11	10
	000				
	001				
	011				
	010				
	100				
	101				
	111				
	110				

$Y_1Y_2Y_3, Z$

DOMANDA N. 3 – Si identifichi l'espressione minima SP per la variabile di stato Y_1 , evitando a priori il pericolo di alee statiche.

	X_1X_2			
y_2y_3	00	01	11	10
00				
01				
11				
10				

$y_1=0$

	X_1X_2			
y_2y_3	00	01	11	10
00				
01				
11				
10				

$y_1=1$

$Y_1 =$

Esercitazione N. 15

L'autoscuola GRATTA&PERDI, per addestrare meglio i suoi allievi, vi chiede di realizzare un simulatore dotato di due ingressi f, c e di una uscita z :

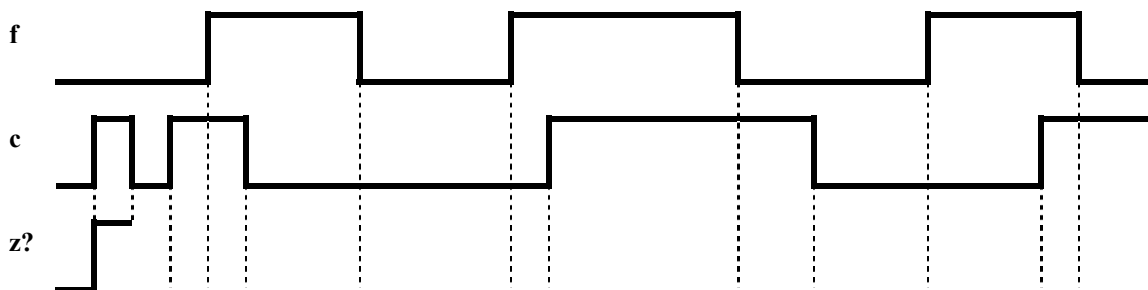
- f è il pedale della frizione (1 premuto, 0 rilasciato),
- c è la leva del cambio (1 marcia inserita, 0 folle),
- z è il comando di un segnalatore acustico (1 rumore di orrenda grattata, 0 nessun suono).

L'allievo può modificare un solo ingresso alla volta.

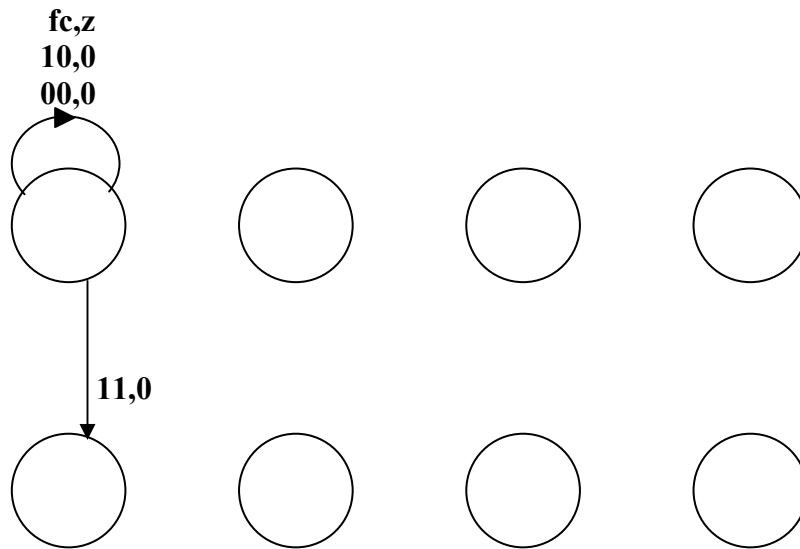
Il rumore della grattata deve essere generato

- a) quando si inserisce una marcia con la frizione non premuta e fino a quando non venga dapprima disinserita la marcia con frizione premuta e poi rilasciata la frizione;
- b) quando si disinserisce una marcia con la frizione non premuta e fino a quando non venga dapprima inserita la marcia con frizione premuta e poi rilasciata la frizione.

DOMANDA N. 1 - Indicare la forma d'onda del segnale d'uscita z in corrispondenza delle forme d'onda dei segnali di ingresso f, c mostrate in figura.



DOMANDA N. 2 – Completare il grafo degli stati.



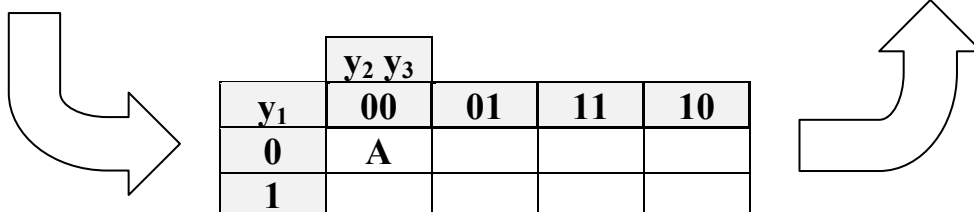
DOMANDA N. 3 - Riempire la tabella di flusso, individuare una corretta codifica degli stati e riempire la tabella delle transizioni.

	fc			
s	00	01	11	10
A				
B				
C				
D				
E				
F				
G				
H				

s^*, z

	fc			
$y_1 y_2 y_3$	00	01	11	10
A:				
B:				
C:				
D:				
E:				
F:				
G:				
H:				

$Y_1 Y_2 Y_3, z$



DOMANDA N. 4 – Riempire le mappe della variabile di stato futuro Y_1 e della variabile d'uscita z ed individuare le “migliori” espressioni SP che le descrivono

		fc			
		00	01	11	10
y ₂ y ₃	00				
	01				
	11				
	10				

$y_1=0$

		fc			
		00	01	11	10
y ₂ y ₃	00				
	01				
	11				
	10				

$y_1=1$

$Y_1 = G(y_1, y_2, y_3, f, c) =$

		fc			
		00	01	11	10
y ₂ y ₃	00				
	01				
	11				
	10				

$y_1=0$

		fc			
		00	01	11	10
y ₂ y ₃	00				
	01				
	11				
	10				

$y_1=1$

$Z = F(y_1, y_2, y_3, f, c) =$