



Sintesi Sequenziale Sincrona

Sintesi Comportamentale di reti Sequenziali Sincrone

Il problema dell'assegnamento degli stati

versione del 9/12/03



Sintesi: Assegnamento degli stati

- La riduzione del numero degli stati minimizza il numero di elementi di memoria e quindi di variabili di stato che descrivono la macchina da sintetizzare
- A pari numero di stati la **complessità della rete combinatoria** che sintetizza la funzione δ **dipende** dal particolare **assegnamento** scelto per gli stati
- L'assegnamento degli stati trasforma la tabella degli stati in **tabella delle transizioni**
- La tabella delle transizioni rappresenta in forma iniziale (manca la scelta del bistabile) l'insieme delle **tabelle delle eccitazioni** della macchina (mappe di Karnaugh soggette ad effettiva sintesi)
- Le **adiacenze** di 1 (o 0) nelle mappe di Karnaugh consentono di ottenere reti combinatorie più o meno complesse, a pari metodo di ottimizzazione

- 2 -



Sintesi: Scelta del codice

- Il processo di codifica degli stati ha l'obiettivo di identificare per ogni rappresentazione simbolica dello stato una corrispondente rappresentazione binaria.
- Due problemi paralleli:
 - Scelta del codice.
 - A minimo numero di bit
 - n° di elementi di memoria = $\lceil \log_2 |S| \rceil$ (codifica densa)
 - One-Hot
 - n° di elementi di memoria = $|S|$ (codifica sparsa)
 - Distanza Minima
 - Gli stati che sono in corrispondenza delle transizioni più frequenti sono poste a distanza Hamming più piccola possibile ponendo il vincolo del minor numero possibile di bit.
 - ...
 - Identificazione della codifica di ogni stato.

- 3 -



Sintesi: Codifica degli stati

- Scelto il codice, la codifica degli stati influisce sia sull'area sia sulle prestazioni del dispositivo.
- Il problema della identificazione della codifica ottima è un problema *NP-completo*
- Impone l'uso di **euristiche per prevedere** l'influenza sul processo di ottimizzazione dell'interazione tra il tipo di elemento di memoria utilizzato e la codifica scelta.
 - Ad esempio, il numero possibili codifiche per il codice a *minimo numero di bit* è:
$$\frac{(2^{\lceil \log_2 |S| \rceil} - 1)!}{(2^{\lceil \log_2 |S| \rceil} - |S|)! \cdot \lceil \log_2 |S| \rceil !}$$
 - Ad esempio, con $|S| = 8$ si hanno 840 possibili codifiche
- Spesso, scelto il codice, si preferisce non ricorrere ad alcuna specifica strategia di codifica.
 - Il *costo della strategia di codifica rispetto alla affidabilità del risultato ottenuto* è ritenuto eccessivo.

- 4 -



Sintesi: Scelta del codice

- Semplici codifiche: binario naturale e one-hot con codifica random
 - Binario Naturale:
 - Il numero di bit è quello minimo
 - al primo stato corrisponde la configurazione di bit associata a 0, al secondo stato corrisponde la configurazione di bit associata ad 1...
 - L'ordinamento degli stati è quello determinato in fase di realizzazione della tabella degli stati.
 - One-Hot:
 - Il numero di bit per la codifica dello stato è pari al numero degli stati
 - In ogni codifica, un solo bit assume valore 1. Tutti i bit rimanenti assumono valore 0
 - Si osservi che le codifiche degli stati sono tutte a distanza di hamming 2

- Esempio:

| | Binario naturale | One-Hot |
|-------|------------------|---------|
| S_0 | 00 | 001 |
| S_1 | 01 | 010 |
| S_2 | 10 | 100 |

- 5 -



Codifica degli stati: **codifica a numero minimo di bit e flip-flop D**

- Consideriamo il caso di codifica a **numero minimo di bit** e utilizzo di **flip flop D**
- E' possibile utilizzare **metodi euristici** per determinare codifiche che possano produrre macchine con reti combinatorie semplificate da una **buona scelta** dell'associazione **codifica-stato**
- Nel caso di bistabili D è possibile identificare dei **criteri di scelta semplici**, poichè la **tabella delle transizioni** della macchina **coincide** con la **tabella delle eccitazioni**
- I criteri di scelta si basano sul principio di **generare** il più possibile **1** (o **0**) **adiacenti** nella tabella delle transizioni (eccitazioni)

- 6 -



Codifica degli stati: **codifica a numero minimo di bit e flip-flop D**

- Uno dei metodi utilizzabili manualmente, su macchine con un numero di stati ridotto, si basa sulle seguenti considerazioni che generano **vincoli** di codifica:
 1. Se **due stati** s_i e s_j hanno, per la stessa configurazione di ingresso, lo **stesso stato futuro** è opportuno che s_i e s_j abbiano **codifiche adiacenti**, in modo da avere coppie di 1 o di 0 **adiacenti sulle colonne**
- **Esempio - punto 1.**

| | 00 | 01 | 11 | 10 | Z |
|----|----|----|----|----|---|
| S0 | S0 | S0 | S2 | S1 | 1 |
| S1 | S1 | S1 | S0 | S1 | 0 |
| S2 | S2 | S3 | S0 | S2 | 1 |
| S3 | S3 | S3 | S2 | S3 | 0 |

| | |
|----|----|
| S0 | 00 |
| S1 | 01 |
| S2 | 11 |
| S3 | 10 |

| | 00 | 01 | 11 | 10 | Z |
|----|----|----|----|----|---|
| 00 | 00 | 00 | 11 | 01 | 1 |
| 01 | 01 | 01 | 00 | 01 | 0 |
| 11 | 11 | 10 | 00 | 11 | 1 |
| 10 | 10 | 10 | 11 | 10 | 0 |

- 7 -



Codifica degli stati: **codifica a numero minimo di bit e flip-flop D (cont.)**

2. Se **due stati** s_i e s_j sono **stati prossimi** dello stesso stato e **corrispondono a ingressi adiacenti**, è opportuno che abbiano **codifiche adiacenti**, in modo da avere coppie di 1 o di 0 **adiacenti sulle righe**
 3. Nel caso di **macchina di Mealy** è possibile esprimere un criterio anche relativo all'uscita (se s_i e s_j hanno **uscite identiche**, per qualche ingresso, è opportuno che i due stati abbiano **codifiche adiacenti**)
- **Esempio - punto 2.**

| | 00 | 01 | 11 | 10 | Z |
|----|----|----|----|----|---|
| S0 | S0 | S0 | S2 | S1 | 1 |
| S1 | S1 | S1 | S0 | S1 | 0 |
| S2 | S2 | S3 | S0 | S2 | 1 |
| S3 | S3 | S3 | S2 | S3 | 0 |

| | |
|----|----|
| S0 | 00 |
| S1 | 01 |
| S2 | 11 |
| S3 | 10 |

| | 00 | 01 | 11 | 10 | Z |
|----|----|----|----|----|---|
| 00 | 00 | 00 | 11 | 01 | 1 |
| 01 | 01 | 01 | 00 | 01 | 0 |
| 11 | 11 | 10 | 00 | 11 | 1 |
| 10 | 10 | 10 | 11 | 10 | 0 |

- 8 -



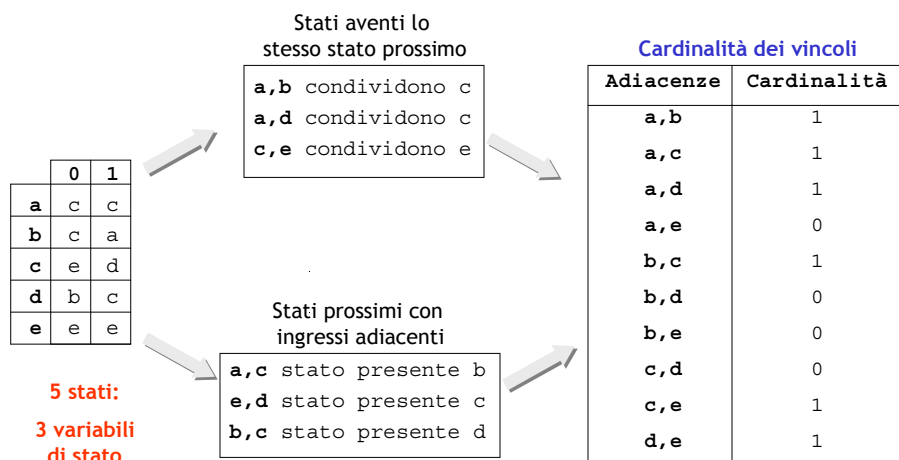
Codifica degli stati: **codifica a numero minimo di bit e flip-flop D** (cont.)

- I vincoli imposti dai tre criteri di adiacenza possono generare conflitti e comunque può risultare impossibile soddisfarli
- A questi vincoli può essere associata una **priorità** e un **peso** relativo:
 - **priorità**: regola 1., priorità max, regola 2., priorità media, regola 3., priorità min
 - **peso**: cardinalità del vincolo derivante dall'esame della tabella degli stati, dopo aver applicato le regole esposte
- Noi lavoriamo considerando solo le regole **1.** e **2.** e il **peso**

- 9 -



Codifica degli stati: **Esempio 1**



- 10 -



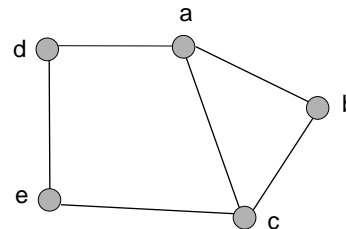
Codifica degli stati: *Esempio 1*

Cardinalità dei vincoli

| Adiacenze | Cardinalità |
|-----------|-------------|
| a, b | 1 |
| a, c | 1 |
| a, d | 1 |
| a, e | 0 |
| b, c | 1 |
| b, d | 0 |
| b, e | 0 |
| c, d | 0 |
| c, e | 1 |
| d, e | 1 |

Dalla tabella dei vincoli, si costruisce il grafo, i cui archi hanno un peso pari alla cardinalità dei vincoli. Il peso viene usato se non è possibile soddisfare tutti i vincoli

Grafo dei vincoli con archi pesati

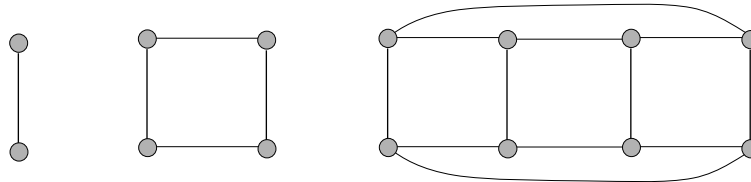


- 11 -



Scelta dell'assegnamento

- Il grafo dei vincoli, il linea di principio, rappresenta l'insieme dei **sotto-cubi di adiacenza** che devono essere riportati nella **mappa di codifica**.
- Ciò è possibile solo se il grafo ottenuto è costituito da soli **n-cubi** o da **unioni di sottocubi**

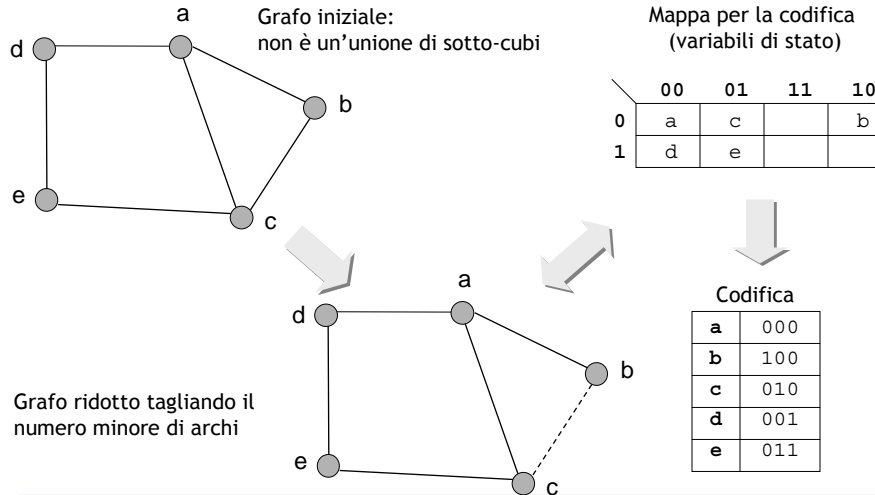


- Se il grafo non è costituito da soli n-cubi o da unioni di n-cubi, è necessario **"tagliare" alcuni archi**
- La scelta viene fatta **eliminando il minimo numero** di archi possibile e utilizzando il **peso come criterio secondario**

- 12 -



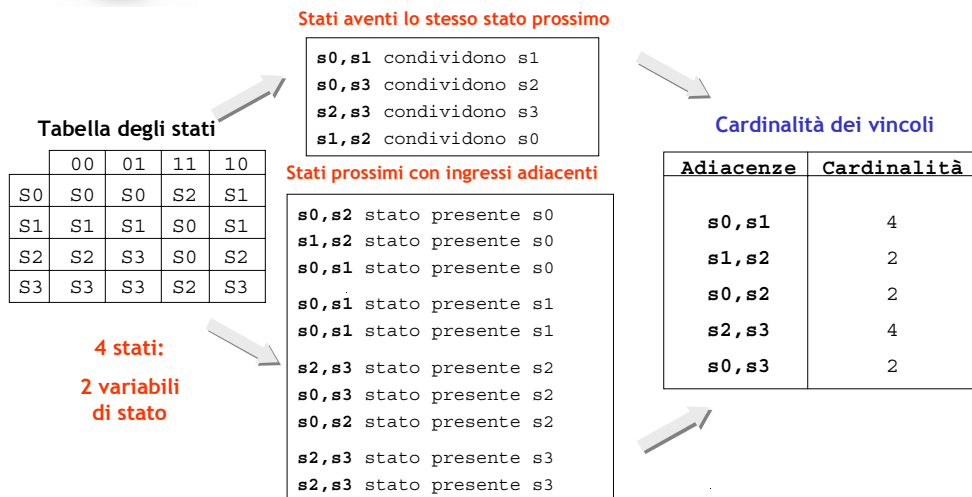
Codifica degli stati: *Esempio 1*



- 13 -



Codifica degli stati: *Esempio 2*



- 14 -



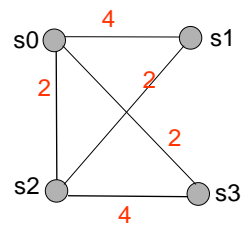
Codifica degli stati: *Esempio 2*

Cardinalità dei vincoli

| Adiacenze | Cardinalità |
|-----------|-------------|
| s0, s1 | 4 |
| s1, s2 | 2 |
| s0, s2 | 2 |
| s2, s3 | 4 |
| s3, s0 | 2 |

Dalla tabella dei vincoli, si costruisce il grafo, i cui archi hanno un peso pari alla cardinalità dei vincoli. Il peso viene usato se non è possibile soddisfare tutti i vincoli

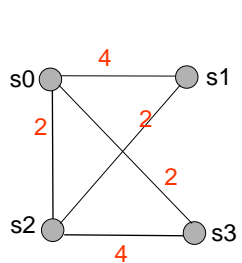
Grafo dei vincoli con archi pesati



- 15 -



Codifica degli stati: *Esempio 2*

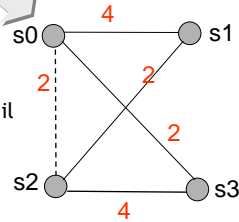


Grafo iniziale:
non è un'unione di sotto-cubi

Mapa per la codifica
(variabili di stato)

| | 0 | 1 |
|---|----|----|
| 0 | s0 | s1 |
| 1 | s3 | s2 |

Grafo ridotto tagliando il
numero minore di archi



Codifica

| | |
|----|----|
| s0 | 00 |
| s1 | 01 |
| s2 | 11 |
| s3 | 10 |

- 16 -



Altri criteri di assegnamento

- Esistono metodi che hanno lo scopo di individuare per la funzione δ una **dipendenza ridotta dalle variabili di stato**
- Il metodo assegna, se possibile, gli stati in modo da identificare dei moduli che dipendono da un numero di variabili di stato inferiore a quello totale della FSM
- Il metodo **partiziona le variabili di stato** (i bistabili) e quindi individua dei moduli costituiti da
 - un sottoinsieme di bistabili e una rete combinatoria che realizza δ solo per quel sottoinsieme di variabili di stato
 - le reti combinatorie risultanti sono in generale localmente meno complesse perché dipendono, oltre che dagli ingressi, da un numero ridotto di variabili di stato

- 17 -



Strumenti di sintesi automatica

- Esistono strumenti specifici di sintesi automatica in grado di adottare diverse strategie per definire la codifica degli stati
- In generale utilizzano le regole esposte per identificare i vincoli di adiacenza tra stati. Il criterio di soddisfacimento dei vincoli dipende dalla strategia adottata che, spesso, può essere definita al momento dell'attivazione dello strumento
- Gli strumenti sono in grado di gestire un numero sufficientemente elevato di variabili nelle tabelle delle verità ottenute dall'assegnamento
- Presuppongono di lavorare con Flip-Flop D e il risultato dell'assegnamento e della relativa ottimizzazione della rete combinatoria dipende anche dalla tecnologia implementativa prevista (es. PLA)

- 18 -