# Decomposizione di funzioni

Decomposizione semplice a 4 variabili di cui 3 al contorno ed 1 indipendente:

Si costruisca una funzione che sia “sicuramente” decomponibile nella forma scelta, si calcolino i termini minimi che la compongono. e successivamente se ne verifichi la corretta decomposizione attraverso le mappe di decomposizione.

Partiamo dalla funzione

*d*

Che evidentemente è decomponibile nella forma desiderata e successivamente si sostituisca a f una funzione CHE non sia a sua volta decomponibile (altrimenti alla fine si andrebbe incontro ad una forma di decomposizione iterativa. Si noti che tutte le funzioni elementari sono a loro volta decomponibili quali :

* AND (a,b,c)
* OR (a,b,c)
* XOR (a,b,c)
* NOR (a,b,c)
* NAND (a,b,c)
* AOI (a,b,c)
* …

Si deve quindi ricorrere a una funzione a 3 variabili più complessa, quale ad esempio:

Sostituendola nell’equazione di cui sopra

Che può essere sviluppata in

Che nella prima forma canonica è

Organizzando gli implicanti, ordinando i litterali in base alla numerazione binaria ed assumendo ‘a’ come MSB e ‘d’ come LSB essi sono composti rispettivamente dai seguenti termini minimi:

:

Quindi la funzione è composta dai seguenti termini minimi:

{1,3,4,6,8,11,12,15}

A questo punto si ipotizzi di NON sapere nulla dell’origine della funzione e di come essa sia stata generata e la si utilizzi come punto di partenza per verificare se vi si possa riconoscere una qualche forma di decomposizione.

Si verifichi ora quali siano le sue possibili decomposizioni.

Si riportino i termini minimi sulle opportune mappe di decomposizione

variabile indip.(verticale) x1

'000' '001' '011' '010' '110' '111' '101' '100'

'0' . 1 1 . 1 . . 1

'1' 1 . 1 . 1 . 1

variabile indip.(verticale) x2

'000' '001' '011' '010' '110' '111' '101' '100'

'0' . 1 1 . . 1 . 1

'1' 1 . . 1 1 . 1

variabile indip.(verticale) x3

'000' '001' '011' '010' '110' '111' '101' '100'

'0' . 1 . 1 1 . . 1

'1' . 1 . 1 1 1 .

variabile indip.(verticale) x4

'000' '001' '011' '010' '110' '111' '101' '100'

'0' . . 1 1 1 . 1

'1' 1 1 . . . 1 1 .

variabili indip. (verticali) x1 e x2

'00' '01' '11' '10'

'00' . 1 1 .

'01' 1 . . 1

'11' 1 . 1

'10' 1 . 1 .

variabili indip. (verticali) x1 e x3

'00' '01' '11' '10'

'00' . 1 . 1

'01' . 1 . 1

'11' . 1 1

'10' 1 . . 1

variabili indip. (verticali) x1 e x4

'00' '01' '11' '10'

'00' . . 1 1

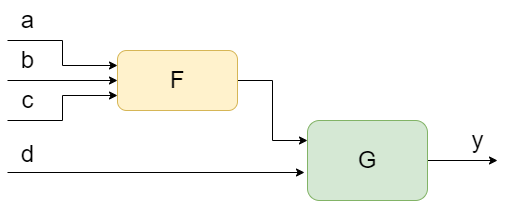
'01' 1 1 . .

'11' . 1 1 .

'10' 1 . 1

Una attenta analisi porta a concludere che la sola mappa che suggerisca la presenza di una decomposizione è quella marcata in rosso dove la prima riga equivale alla seconda negata, che vede “d” come variabile indipendente e “a,b,c” come variabili al contorno.

La decomposizione seguirà pertanto del tipo qui sotto riportato:



Per individuare le funzioni F e G Coinvolte si prenda in esame la mappa di decomposizione che vede la 4^a variabile (“d”) come variabile indipendente: la funzione che lega y alle variabili di ingresso può essere descritta da questa mappa:



Supponendo ad esempio di definire come F=F(a,b,c) la funzione rappresentata dalla seconda riga, giocoforza la prima riga è rappresentativa della stessa funzione negata (not(F)).

Si può pertanto sostituire nella definizione della funzione y, alle tre variabili al contorno (a,b,c) la funzione che le combina F(a,b,c).



Si noti che come sopra la seconda riga rappresenta F e la prima not(F).

In forma discorsiva si potrebbe esprimere il funzionamento del sistema attraverso la seguente affermazione: *“quando d == 1 l’uscita y coincide con la funzione F, mentre se d==0 l’uscita y sarà rappresentata dalla funzione not(F)”*

In forma logica la funzione può pertanto essere espressa come:

Ecco quindi completamente identificata la funzione G ovvero il blocco che la realizza.

Per quanto riguarda la funzione F, di essa conosciamo la tavola di verità (espressa come già affermato, nella seconda riga della mappa di decomposizione).

|  |  |
| --- | --- |
| a b c | F |
| 000 | 1 |
| 001 | 1 |
| 011 | 0 |
| 010 | 0 |
| 100 | 0 |
| 101 | 1 |
| 111 | 1 |
| 110 | 0 |

Che per comodità rappresentativa può essere organizzata in una mappa di Karnaught

|  |  |  |
| --- | --- | --- |
| bc\a | 0 | 1 |
| 00 | 1 | 0 |
| 01 | 1 | 1 |
| 11 | 0 | 1 |
| 10 | 0 | 0 |

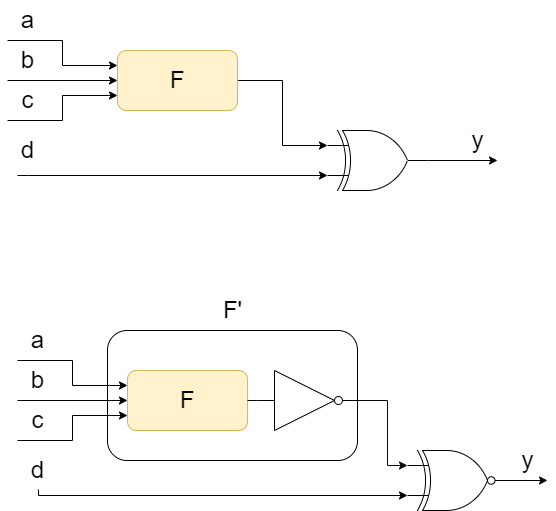
Che attraverso gli implicanti evidenziati suggerisce una rappresentazione nella prima forma canonica come

Confrontando il risultato ottenuto col punto di partenza si potrebbe obiettare che essi sembrano essere diversi, originariamente infatti le funzioni coinvolte erano

Ma si può facilmente constare che F’ = not(F) e che G’ = not(G):

Dove il termine evidenziato è ridondante per il teorema dell’assorbimento

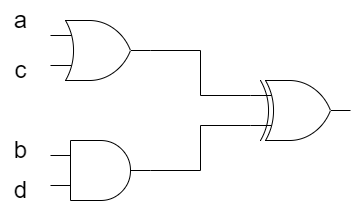
Anche da un punto di vista prettamente grafico si nota la perfetta equivalenza dei due circuiti



Decomposizione Multipla a 4 variabili:

Si costruisca una funzione che sia “sicuramente” decomponibile nella forma scelta, si calcolino i termini minimi che la compongono. e successivamente se ne verifichi la corretta decomposizione attraverso le mappe di decomposizione.

Partiamo ad esempio dalla funzione rappresentata graficamente qui sotto che evidentemente rispetta le specifiche richieste:



Esprimibile come

Che con qualche passaggio ci porta ad evidenziare i seguenti implicanti

Organizzando i termini minimi, ordinando i litterali in base alla numerazione binaria ed assumendo ‘a’ come MSB e ‘d’ come LSB, i vari implicati sono composti rispettivamente dai seguenti termini minimi:

Quindi, riassumendo la funzione è composta dai seguenti termini minimi:

{2,3,5,6,8,9,10,11,12,14}

Partendo dai termini minimi rappresentativi della funzione si vada ora a verificare per esercizio quale tipo di decomposizione vi si riconosce: Inserendo questi termini minimi nelle mappe di decomposizione:

variabile indip.(verticale) x1

'000' '001' '011' '010' '110' '111' '101' '100'

'0' . . 1 1 1 . 1 .

'1' 1 1 1 1 1 . . 1

variabile indip.(verticale) x2

'000' '001' '011' '010' '110' '111' '101' '100'

'0' . . 1 1 1 1 1 1

'1' . 1 . 1 1 . . 1

variabile indip.(verticale) x3

'000' '001' '011' '010' '110' '111' '101' '100'

'0' . . 1 . 1 . 1 1

'1' 1 1 . 1 1 . 1 1

variabile indip.(verticale) x4

'000' '001' '011' '010' '110' '111' '101' '100'

'0' . 1 1 . 1 1 1 1

'1' . 1 . 1 . . 1 1

variabili indip. (verticali) x1 e x2

'00' '01' '11' '10'

'00' . . 1 1

'01' . 1 . 1

'11' 1 . . 1

'10' 1 1 1 1

variabili indip. (verticali) x1 e x3

'00' '01' '11' '10'

'00' . . 1 .

'01' 1 1 . 1

'11' 1 1 . 1

'10' 1 1 . 1

variabili indip. (verticali) x1 e x4

'00' '01' '11' '10'

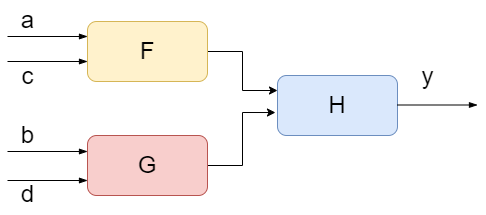
'00' . 1 1 .

'01' . 1 . 1

'11' 1 1 . .

'10' 1 1 1 1

Si evince che l’unica mappa in cui vi sia molteplicità di riga (o di colonna) inferiore a 2 è quella per la quale si sono utilizzate quali variabili indipendenti la 1^a e la 3^a. Inoltre si può notare che per questa specifica mappa sia la molteplicità di riga che quella di colonna risulta inferiore a due. Indice certo che siamo in presenza di due differenti decomposizioni semplici di cui una con variabili indipendenti (la 1 e la 3) e l’altra quella relativa alla mappa trasposta che avrebbe variabili indipendenti la (2 e la 4).  
  
La presenza di entrambe le decomposizioni per il ben noto teorema suggerisce che ci si trova di fronte ad una decomposizione multipla del tipo riportato in figura:



Le varie sotto-funzioni che compongono lo schema appena visto possono essere ricavate direttamente dalla tavola di decomposizione:

ac\bd '00' '01' '11' '10'

'00' 0 0 1 0

'01' 1 1 0 1

'11' 1 1 0 1

'10' 1 1 0 1

Le righe della tabella rappresentano G e not(G) mentre le righe rappresentano F e not(F). Poi è assolutamente arbitrario scegliere se G sia rappresentata dalla prima riga e not(G) dalle altre tre o viceversa. (Analogo discorso per quanto riguarda F.

Supponiamo ad esempio di scegliere come G(bd) la funzione con la tabella di verità esplicitata nelle righe 2,3 e 4.

|  |  |
| --- | --- |
| bd | G |
| 00 | 1 |
| 01 | 1 |
| 11 | 0 |
| 10 | 1 |

Risulta evidente che questa funzione può essere espressa come

Si può quindi sviluppare la prima decomposizione semplice che vede il risultato y ottenibile come una funzione esclusivamente delle variabili a,c,G (y = y (a,c,G) ) esprimibile secondo la seguente mappa (estratta direttamente dalla mappa di decomposizione)

|  |  |  |
| --- | --- | --- |
| ac\G | 0 | 1 |
| 00 | 1 | 0 |
| 01 | 0 | 1 |
| 11 | 0 | 1 |
| 10 | 0 | 1 |

Però per quanto visto sopra questa funzione deve risultare a sua volta decomponibile. Infatti le colonne della tabella possono essere esplicitate nelle funzioni F e not(F)

Supponendo di scegliere come F la prima colonna e come not(F) la seconda la funzione in esame avrà la seguente tavola di verità:

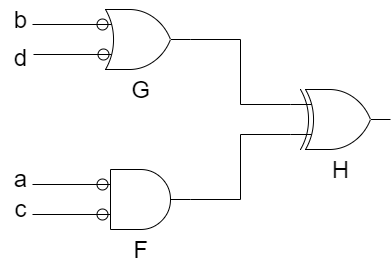
|  |  |
| --- | --- |
| ac | F |
| 00 | 1 |
| 01 | 0 |
| 11 | 0 |
| 10 | 0 |

Ovvero:

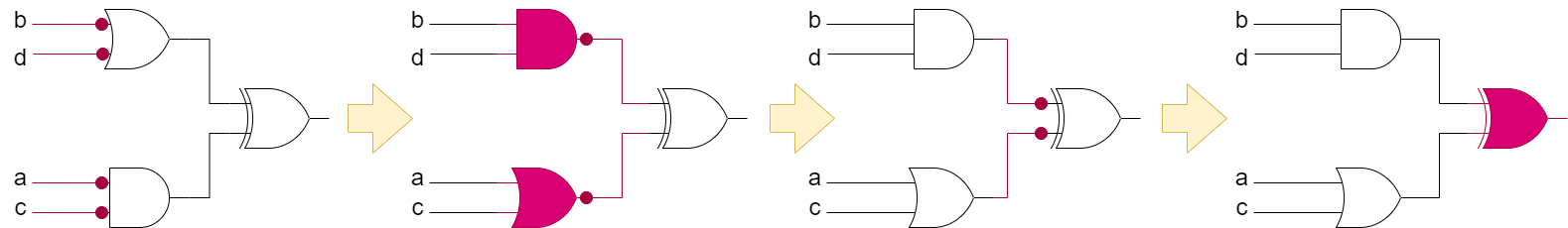
Attuando quindi questa ulteriore sostituzione (di F all’interno della tabella di y=y(a,c,G) ) si riesce ad evidenziare y = H (F,G)

|  |  |  |
| --- | --- | --- |
| F\G | 0 | 1 |
| 0 | 0 | 1 |
| 1 | 1 | 0 |

Da cui risulta evidente che la funzione H è uquale a XOR(F,G) e quindi lo schema finale ricavato è il seguente



Si potrebbe obiettare come nell’esercizio precedente, che esso non sia esattamente uguale alla funzione dalla quale si è partiti, ma sfruttando il teorema di De Morgan e “spingendo in avanti le bolle” si possono ottenere le seguenti modifiche al circuito:

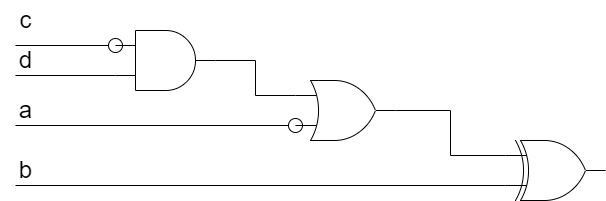


Che dimostrano come il circuito ottenuto abbia la medesima funzione di trasferimento del circuito di partenza, sebbene realizzato con porte logiche diverse.

Decomposizione Iterativa a 4 variabili:

Si costruisca una funzione che sia “sicuramente” decomponibile nella forma scelta, si calcolino i termini minimi che la compongono. e successivamente se ne verifichi la corretta decomposizione attraverso le mappe di decomposizione.

Partiamo ad esempio dalla funzione rappresentata graficamente qui sotto che evidentemente rispetta le specifiche richieste:



Il cui funzionamento può essere espresso dalla seguente equazione:

E quindi dai seguenti implicanti

Ora organizzando i termini minimi in forma numerica secondo la numerazione binaria organizzando le variabili in maniera ordinata dove “a” rappresenta il bit più significativo e “d” il meno significativo si trovano i seguenti termini minimi:

Quindi concludendo la prima parte la funzione in esame è composta dai seguenti termini minimi:

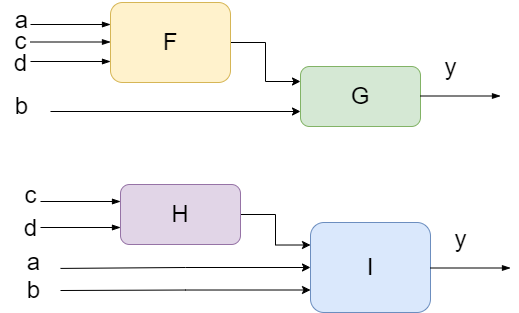
{0,1,2,3,9,12,14,15}

Vediamo adesso, partendo direttamente dai termini minimi e supponendo di dimenticarci di come essi siano stati ricavati, come le mappe di decomposizione ci possono aiutare a riconoscere la presenza e la tipologia di decomposizione.

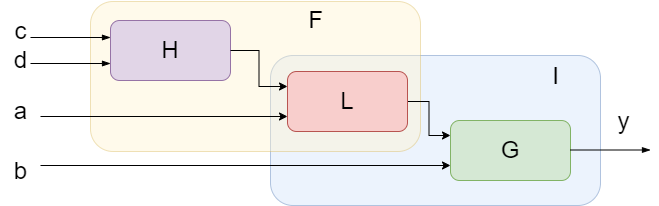


Dall’analisi delle mappe si potrà riconoscere che quella che utilizza come variabile indipendente la seconda (x2) presenta una molteplicità di colonna pari a 2. Infatti la prima riga è la negata della seconda.  
Inoltre anche la mappa che adotta come variabili indipendenti la prima e la seconda (x1 e X2) presenta molteplicità di colonna pari a due (vi sono solo 2 colonne distinte 1010 e 1001). Tutte le altre mappe non evidenziano alcunché di particolare.

Già da questa semplice analisi si può intuire che la funzione è decomponibile in due forme semplici qui di seguito riportate:



Inoltre il fatto che tali decomposizioni debbano sussistere contemporaneamente nello stesso circuito ci fa intuire che le funzioni F ed I debbano giocoforza essere a loro volta nuovamente decomponibili suggerendoci quindi la struttura qui sotto rappresentata:



Da cui si evince che il blocco H deve essere contenuto all’interno del blocco F ed al contempo il blocco G deve essere parte integrante del blocco I. Contemporaneamente si evince la presenza del blocco L che in realtà è condiviso tra il blocchi F ed I ottenuti nelle decomposizioni semplici.

Lo scopo finale della semplificazione è quello di trovare i tre blocchi L, G, H che dovrebbero rappresentare la realizzazione più economica in termini di area occupata dal circuito.

Mentre i blocchi H e G possono essere dedotti facilmente dalle due decomposizioni semplici rilevate precedentemente, il riconoscimento del blocco H può essere fatto secondo due strade distinte: ovvero

* o evidenziando una decomposizione ulteriore all’interno del blocco F
* o evidenziando una decomposizione ulteriore all’interno del blocco I

In questo esercizio seguiremo entrambe le strade e vedremo che conducono al medesimo risultato ma è evidente che per uno svolgimento corretto basta seguirne una sola.

Decomposizione con 1 variabile indipendente (Blocchi F e G)

Si prenda in considerazione la seconda tavola di decomposizione:



Si può notare che le due righe sono una il complemento dell’altra. Ogni riga rappresenta una funzione di tre variabili (rispettivamente 1,3,4 – ovvero per la nomenclatura usata a,c,d) mentre sull’asse verticale vi è la variabile b.

Se si sostituisce alle 3 variabili al contorno a,c,d la funzione F stessa che le “racchiude” si può riscrivere la tabella di cui sopra nella forma:



Si può evidenziare che il blocco G = G(b,F) è realizzabile attraverso una funzione XNOR

L’analisi della funzione F e della sua ulteriore decomposizione la lasciamo a dopo, ma si volendo si potrebbe già anticipare che secondo la tavola di verità espressa dalla seconda riga della tavola di decomposizione

Dove risulta evidente che la variabile ‘a’ potrebbe essere raccolta a fattore comune evidenziando quindi una possibile decomposizione completando di conseguenza l’esercizio.

Decomposizione con 2 variabile indipendente (Blocchi H e I)

Si prenda in considerazione la tabella che vede come variabili indipendenti la 1 e la 2 (per noi ‘a’ e ‘b')

E come variabili al contorno ‘c’ e ‘d’



Si può notare come la molteplicità di colonna sia pari a 2 ovvero, letta per righe:

* la prima riga rappresenta la funzione “sempre 1”
* la seconda riga rappresenta la funzione “sempre 0”
* la terza e la quarta sono l’una la negazione dell’altra per cui le definiamo come ‘H’ e ‘not(H)’

Sfruttando la tabella di verità rappresentata nella quarta riga della tavola di decomposizione si evince che:

Con questa semplificazione, ovvero sostituendo alle variabili ‘c’ e ‘d’ la funzione ‘H’ che le raccoglie, il sistema può essere rappresentato attraverso la seguente mappa di Karnaugh che rappresenta il blocco I



(che sarà a sua volta decomponibile)

Ricerca del blocco L all’interno del blocco F

Si sa che la funzione F deve essere a sua volta decomponibile usando ‘a’ come variabile indipendente, quindi si può organizzare una sua mappa di decomposizione che evidenzi questa prerogativa:



La decomposizione risulta evidente dal fatto che la prima riga è composta da soli zeri mentre la seconda può essere ricondotta ad una funzione esclusivamente di c e d.

Adesso vi sono due possibilità:

* riconoscere in tale funzione la funzione H già definita in precedenza
* ridefinire ex-novo una funzione di (c,d)

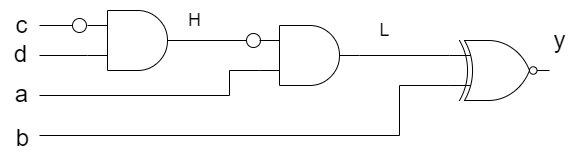
1. Nel primo caso possiamo ricordare che H è stata individuata nella funzione

Quindi risulta evidente che la seconda riga della tabella di cui sopra rappresenta not(H), pertanto la tabella può essere riformulata come:



E quindi la funzione L=L(a,H) può essere espressa con

E quindi lo schema finale risulta:



1. Nel secondo caso possiamo seguire una strada un po’ diversa: individuare una nuova funzione di (c,d) nella seconda riga (che chiameremo H’) e che è esprimibile come

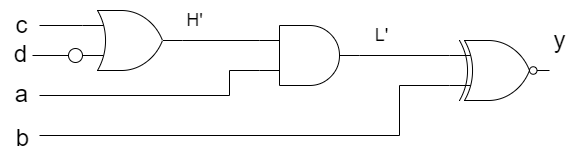
(si noti che la funzione alla quale si è pervenuti altri non è se non “not(H)”)

E ridefinire la funzione L, che per congruità chiameremo L’, e che è esprimibile secondo :



Da cui

Lo schema riassuntivo è pertanto:



Risulta evidente che entrambi i percorsi hanno portato a risultati simili:

nel primo caso abbiamo identificato la decomposizione

nel secondo

Che, sebbene realizzabili utilizzando porte logiche diverse hanno evidentemente la medesima funzione (v. Terorema di De Morgan)

Ricerca del blocco L all’interno del blocco I

A puro scopo di esercizio vediamo che una strada alternativa può essere quella di cercare una ulteriore decomposizione partendo del blocco I invece che dal blocco F.

Partendo da quanto già trovato, ovvero:

Ricordiamo che il blocco I è esprimibile secondo la seguente tabella



e si sa dalle considerazioni iniziali che esso deve essere decomponibile usando la variabile ‘*b’* come variabile indipendente. Si riorganizzi pertanto la tabella di cui sopra per evidenziare tale funzionamento:



Anche in questo caso si potrebbe scegliere se la funzione L=L(a,H) sia rappresentata dalla seconda riga o dalla prima, fermo restando che evidentemente le due righe rappresentano una funzione e la sua negazione.

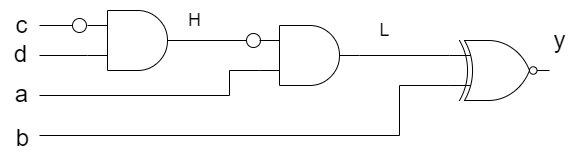
* Nel primo caso veniamo a definire L come:

Che coincide a quanto fatto sopra e la funzione G che lega L con b è esprimibile come



Ovvero è una funzione XNOR (che coincide col blocco G sopra trovato)

E lo schema risulta quindi identico a quello trovato sopra:



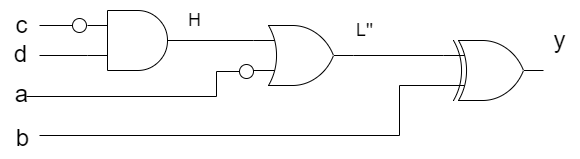
* Nel secondo caso veniamo a definire L’’ come:

Il che ci porta a ri-definire la funzione G’’ che lega L con b attraverso la tabella



E quindi ad identificare la stessa con una funzione XOR.

Il risultato finale risulta essere quindi.



Si noti che a seconda di quali scelte si operano nella definizione delle sotto-funzioni si possono ottenere risultati che anche se realizzabili con porte logiche diverse hanno comunque il medesimo funzionamento.

Mappe di decomposizione “custom”

Scrivere ma mappa di decomposizione in 5 variabili che veda x1 e x4 come variabili indipendenti e x2,x3,x5 quali variabili al contorno.

Premesso che evidentemente ciascuna mappa di decomposizione può essere sempre letta sia per righe che per colonne scambiando di fatto il gruppo delle variabili al contorno con quello delle variabili indipendenti, si inizi col definire gli “assi” della matrice e le combinazioni tra le variabili su di essi.

Inoltre poiché le mappe di decomposizione vengono utilizzate fondamentalmente per individuare “sotto-funzioni” comuni, si è assolutamente liberi di utilizzare l’ordinamento nelle combinazioni tra variabili più opportuno, anche laddove questo portasse a mappare su celle distanti temini minimi con distanza unitaria uno dall’altro.

L’ordinamento in binario è di certo più agevole rispetto l’ordinamento secondo Gray così come pure rispetto altri ordinamenti che ad esempio fondano in qualche modo codifiche diverse (ad esempio gray e binario).



Si passi poi ad individuare all’interno di essa le posizioni dei termini minimi m0 e quelli rappresentabili con tutte le potenze di 2.



Si può quindi completare la matrice prendendo:

* Il pattern (0 ..1) e seguendone l’andamento su (2..3)
* Il pattern (0 ..3) e seguendone l’andamento su (4..7)
* Il pattern (0 ..7) e seguendone l’andamento su (8..16)
* Il pattern (0 ..16) e seguendone l’andamento su (17..32)









Da ultimo, si suggerisce di effettuare una verifica:  
Si prenda qualche numero a caso che non presentino particolari simmetrie nella codifica binaria e si verifichi il loro corretto posizionamento all’intersezione delle rispettive coordinate

Ad esempio:

9 = 01001 🡪 x1,x4=0,0 x2,x3,x5=101

20 = 10100 🡪 x1,x4=1,0 x2,x3,x5=010

23 = 10111 🡪 x1,x4=1,1 x2,x3,x5=011

