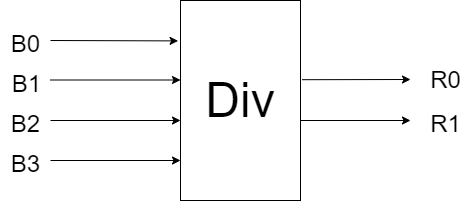
# Funzioni Simmetriche

Riconoscimeto e Realizzazione:

Si consideri un numero intero, rappresentato in binario nel formato unsigned a 4 bit.  
Si realizzi un dispositivo in grado di fornire il resto della divisione per 3 del numero stesso.

Si dentifichino i segnali coinvolti:

Il circuito definitivo dovrà presentare 4 bit di ingresso [B3..B0]che rappresentano i numeri binari in ingresso compresi tra 0 e 15. Dove B3 è il MSB e B0 è il LSB e 2 bit di uscita [R1,R0] che rappresenteranno il resto della divisione per 3.



Si esprima attraverso la tavola di verità il rapporto tra ingressi ed uscita del circuito:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N | resto | B3 | B2 | B1 | B0 | R1 | R0 |
|  |  |  |  |  |  |  |  |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 |
| 2 | 2 | 0 | 0 | 1 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 4 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 5 | 2 | 0 | 1 | 0 | 1 | 1 | 0 |
| 6 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |
| 7 | 1 | 0 | 1 | 1 | 1 | 0 | 1 |
| 8 | 2 | 1 | 0 | 0 | 0 | 1 | 0 |
| 9 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |
| 10 | 1 | 1 | 0 | 1 | 0 | 0 | 1 |
| 11 | 2 | 1 | 0 | 1 | 1 | 1 | 0 |
| 12 | 0 | 1 | 1 | 0 | 0 | 0 | 0 |
| 13 | 1 | 1 | 1 | 0 | 1 | 0 | 1 |
| 14 | 2 | 1 | 1 | 1 | 0 | 1 | 0 |
| 15 | 0 | 1 | 1 | 1 | 1 | 0 | 0 |

Si evidenzino i termini minimi che compongono rispettivamente le funzioni R0 ed R1

|  |  |  |  |
| --- | --- | --- | --- |
| R1 |  |  |  |
| B3 | B2 | B1 | B0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 |
|  |  |  |  |
|  |  |  |  |
| R0 |  |  |  |
| B3 | B2 | B1 | B0 |
| 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Si passi ora a verificare se le funzioni siano simmetriche: si calcoli per ciascun colonna il rapporto tra uni e zeri:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| R1 |  |  |  |  | R0 |  |  |  |
| 0 | 0 | 1 | 0 |  | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |  | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 |  | 0 | 1 | 1 | 1 |
| 1 | 0 | 1 | 1 |  | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 |  | 1 | 1 | 0 | 1 |
| 3/2 | 2/3 | 3/2 | 2/3 |  | 2/3 | 3/2 | 2/3 | 3/2 |

Si noti come compaiano per ogni tabella solo 2 valori (2/3 e 3/2) di cui uno il reciproco dell’altro. Si può pertanto provare ad invertire le colonne corrispondenti ad un certo valore. Ad esempio si invertano le colonne 1 e 3 tranto nella prima che nella seconda tabella.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| R1 |  |  |  | liv |  | R0 |  |  |  | liv |
| 1 | 0 | 0 | 0 | 1 |  | 1 | 0 | 1 | 1 | 3 |
| 1 | 1 | 1 | 1 | 4 |  | 1 | 1 | 1 | 0 | 3 |
| 0 | 0 | 1 | 0 | 1 |  | 1 | 1 | 0 | 1 | 3 |
| 0 | 0 | 0 | 1 | 1 |  | 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 0 | 1 |  | 0 | 1 | 1 | 1 | 3 |
| 2/3 | 2/3 | 2/3 | 2/3 |  |  | 3/2 | 3/2 | 3/2 | 3/2 |  |

Così facendo tutte le colonne di ciascuna tabella hanno il medesimo rapporto, mentre i pesi dei vari termini minimi della funzione così modificata si possono raggruppare in 2 gruppi: di peso rispettivamente 1 e 4 per R! e in 3 e 0 per R0.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| R1 |  |  |  | liv |  | R0 |  |  |  | liv |
| 1 | 0 | 0 | 0 | 1 |  | 1 | 0 | 1 | 1 | 3 |
| 0 | 0 | 1 | 0 | 1 |  | 1 | 1 | 1 | 0 | 3 |
| 0 | 0 | 0 | 1 | 1 |  | 1 | 1 | 0 | 1 | 3 |
| 0 | 1 | 0 | 0 | 1 |  | 0 | 1 | 1 | 1 | 3 |
| 1/3 | 1/3 | 1/3 | 1/3 |  |  | 3/1 | 3/1 | 3/1 | 3/1 |  |
|  |  |  |  | liv |  |  |  |  |  | liv |
| 1 | 1 | 1 | 1 | 4 |  | 0 | 0 | 0 | 0 | 0 |
| inf | inf | inf | inf |  |  | 0 | 0 | 0 | 0 |  |

Poiché il numero dei termini minimi inclusi in ciascun gruppo è congruo con i termini necessari per una funzione simmetrica di un solo livello, Si può desumere di trovarsi in presenza di due funzioni simmetriche:

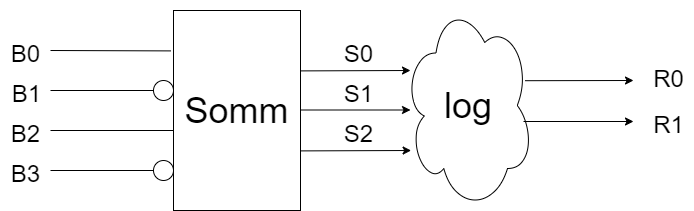
Ad ulteriore riprova si può notare che per ogni sottofunzione (o sottotabella) vale la relazione:

Infatti:

* relativamente alla tabella relativa al livello di simmetria 1 il rapporto è 1/(4-1) =1/3
* relativamente alla tabella relativa al livello di simmetria 4 il rapporto è 4/(4-4) =inf
* relativamente alla tabella relativa al livello di simmetria 3 il rapporto è 3/(4-3) =3/1
* relativamente alla tabella relativa al livello di simmetria 0 il rapporto è 0/(4-0) = 0

Per la realizzazione del circuito ci si può avvalere dell’impiego di sommatori multi-bit ed in particolare, visto che in entrambe le funzioni R1 ed R2 le variabili in ingresso che vanno invertite sono le medesime si può condividere il medesimo sommatore in entrambe le funzioni:

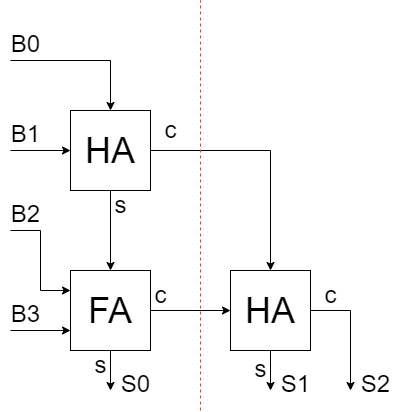
Lo schema di principio sarà il seguente:



Dei quattro bit in ingresso B1 e B3 verranno invertiti e successivamente tutti convergeranno all’interno di un sommatore multibit. Esso eseguirà la somma dei valori assunti dai singoli bit e fornirà in uscita la codifica binaria del peso della parola di ingresso. Poiché il peso sarà compreso tra 0 e 4 esso potrà essere codificato su 3 bit [S2..S0]. Prestare particolare attenzione a definire con chiarezza quale voglia essere il bit più significativo (S2) e quale il meno significativo (S0).

Una volta ottenuto un circuito in grado di stabilire il peso della parola in ingresso, per completare il circuito bisognerà definire una logica opportuna in grado di attivare il bit R1 di uscita se il peso della parola è 1 oppure 4 e al contempo di attivare R0 se il peso è 0 oppure 3.

Il sommatore multibit può essere realizzato sommando prima tutti i bit [B3..B0] attraverso una catena di Full Adder (FA) e Half Adder (HA), a seconda delle necessità e successivamente sommando i riporti da questi generati attraverso un ulteriore catena di sommatori e così via. Nel nostro caso specifico per sommare 4 bit bastano un Half adder che sommi 2 bit B0 e not(B1) (fornendo un bit di somma S ed un bit di riporto C) seguito da un full adder che somma il bit S prelevato dallo stadio precedente con gli ulteriori 2bit della parola di ingresso (B2 e not (B3). Il bit di somma di questo FA fornirà il bit meno significativo della parola finale (S0). IN un secondo stadio bisognerà sommare TUTTI i bit di riporto © generati dal primo stadio. Poiché nel nostro caso specifico il primo stadio ha generato 2 soli bit di riporto, per eseguire la loro somma basterà utilizzare un singolo HA. Il bit di somma da esso generato fornirà il bit S1 della parola finale, mentre il bit di riporto genererà S2, ovvero il bit più significativo della parola finale.



Per definire la Logica che viene a completare il circuito possiamo definire la tabella di verità delle funzioni R0 ed R1 ma questa volta in funzione di [S2..S0].

La logica da sviluppare è riassunta nella tabella sotto riportata:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| peso | S2 | S1 | S0 | R1 | R0 |
| 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 2 | 0 | 1 | 0 | 0 | 0 |
| 3 | 0 | 1 | 1 | 0 | 1 |
| 4 | 1 | 0 | 0 | 1 | 0 |
| - | 1 | 0 | 1 | - | - |
| - | 1 | 1 | 0 | - | - |
| - | 1 | 1 | 1 | - | - |

Si noti in particolare che essendovi solo 4 bit in ingresso al circuito complessivo ([B3..B0]) il peso di questa parola sarà sicuramente compreso tra 0 e 4 e non potrà MAI assumere i valori 5,6 oppure 7. Di conseguenza si sa con certezza che le combinazioni dei bit [S2..S0] che codificano tali valori NON si potranno mai verificare.

Per avere una rappresentazione più chiara degli implicanti che si possono utilizzare per realizzare la logica desiderata si può far ricorso alla mappe di Karnaught. Ancora una volta si sottolinea l’importanza nell’ordine delle variabili S2,S1 ed S0.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| R1 |  |  |  |  |  | R0 |  |  |  |  |
| S2/S1 S0 | 00 | 01 | 11 | 10 |  | S2/S1 S0 | 00 | 01 | 11 | 10 |
| 0 | 0 | 1 | 0 | 0 |  | 0 | 1 | 0 | 1 | 0 |
| 1 | 1 | - | - | - |  | 1 | 0 | - | - | - |

Sfruttando le condizioni non specificate (d.c.) si possono evidenziare le seguenti funzioni logiche:

E rispettivamente

Facilmente realizzabili.

Realizzazione:

**Ci siano 6 interuttori (rappresentati dalle variabili a,b,c,d,e,f) di cui “A” sia l’interruttore “generale” mentre gli altri siano di uguale “importanza” e subordinati a questo.**

**Si progetti un circuito che fornisca un’ uscita pari a 1 se e solo se 3 o più interruttori tra b,c,d,e,f sono accesi ed ‘A’ e’ acceso, mentre l’uscita sia 0 se sono accesi meno di 3 interruttori (tra b,c,d,e,f,) oppure se ‘A’ è spento.**

Già dalla definizione stessa del problema si evince che la funzione che si vuole realizzare sia **parzialmente** **simmetrica**, infatti quando si accenna al fatto che all’interruttore “A” è demandato un certo compito mentre gli altri sono “**di eguale importanza**” si sta sfruttando appunto una delle definizioni che definiscono casa siano le funzioni simmetriche.

Supponendo di racchiudere in un unico blocco che fornisca un uscita “H” la logica che valuta se tre o più interruttori dell’insieme [b,c,d,e,f] siano o meno attivi, il resto della logica può essere evidenziato nella seguente tabella logica:

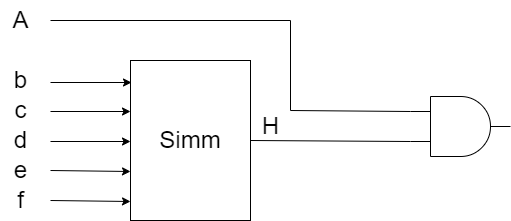
|  |  |  |
| --- | --- | --- |
| H\A | 0 | 1 |
| 0 | 0 | 0 |
| 1 | 0 | 1 |

Dove A è il valore logico assunto dall’interruttore “generale” ed H è l’uscita del blocco (ancora da sviluppare) che deciderà se tre o più interruttori del gruppo [b,c,d,e,f] siano attivi.

La prima colonna esprime il concetto: se A==0 (spento) l’uscita è =0 (spenta)

La seconda colonna esprime il concetto che se A ==1 (acceso) allora l’uscita coinciderà col valore fornito da H stesso.

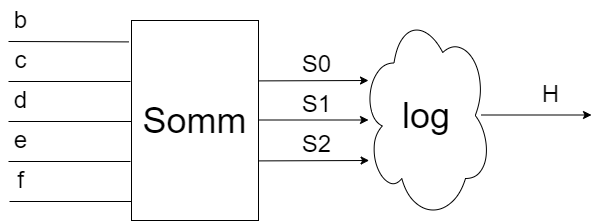
E’ banale notare come tale logica possa essere realizzata da una semplice porta AND. Pertanto l’intero circuito potrà già essere immaginato nel seguente modo:



A questo punto non rimane che descrivere e realizzare la funzione logica H=H(b,c,d,e,f).

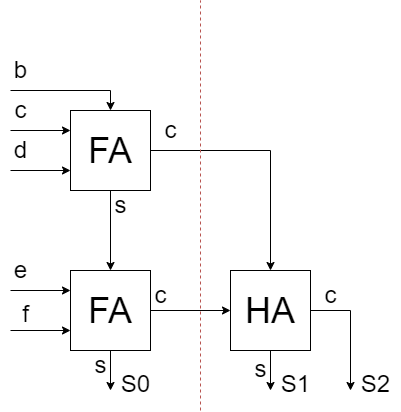
Dalle specifiche del problema si evince che:

Quindi essa può essere realizzata attraverso un sommatore multi-bit che restituisca il valore (espresso in binario) del numero di bit attivi all’ingresso – ovvero il perso della parola [b,c,d,e,f], seguito da una logica che analizzi se tale numero sia o meno maggiore di 3.



Poiché il numero di ingressi è pari a 5 i sommatore restituirà sulle linee [S2..S0] un valore compreso tra 0 e 5 ovvero un numero binario su 3bit compreso tra 000 e 101.

La realizzazione del sommatore può seguire il procedimento a cascata che vede l’utilizzo di 2 Full-Adder per sommare i 5 bit in ingresso ed un ulteriore Half Adder per sommare i riporti da questi generti.



I tre bit di uscita [S2..S0] rappresentano il numero di bit attivi sugli ingressi codificati in binario, dove S2 è il MSB e S0 è il LSB.

A questo punto non rimane che definire la logica che attivi il segnale H se [S2..S0] rappresentano un numero maggiore o uguale a 3.   
La tabella di verità di tale funzione è la seguente:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| peso | S2 | S1 | S0 | R1 |
| 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 2 | 0 | 1 | 0 | 0 |
| 3 | 0 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 1 |
| 5 | 1 | 0 | 1 | 1 |
| - | 1 | 1 | 0 | - |
| - | 1 | 1 | 1 | - |

Si noti in particolare la presenza di valori non definiti per le combinazioni 110 e 111 che NON potranno mai verificarsi all’ingresso.

I termini minimi possono essere organizzati per maggior chiarezza in una mappa di Karnaugh

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S2/S1 S0 | 00 | 01 | 11 | 10 |
| 0 | 0 | 0 | 1 | 0 |
| 1 | 1 | 1 | - | - |

Che suggerisce di adottare quale funzione logica

