

## Q&A: Alcune domande e relative risposte

**Q:** Non mi è del tutto chiaro il metodo di passaggio dalla prima alla seconda forma canonica e viceversa.

Assumendo che nella SOP i mintermini hanno valore 1 e nella POS i Maxtermini valore 0. Per passare da una forma all'altra si dovrebbe negare tutto e basta (utilizzando De Morgan dove  $\sim(xy) = \sim x + \sim y$ ?

Ad esempio:

SOP:  $ABC + A\sim B\sim C$   
che sviluppato viene:  $AC$

POS corrispondente:  
 $\sim(ABC + A\sim B\sim C)$   
che sviluppato viene:  $\sim A + \sim C$

**A:** Magari fosse così semplice....

La SOP si ottiene sommando (OR) tutti i mintermini della funzione cioè tutti i mintermini per cui la funzione vale 1. I mintermini sono funzioni contenenti tutti i letterali, eventualmente negati, che hanno la caratteristica di assumere valore 1 solo per una particolare configurazione per cui anche la funzione di partenza vale 1. Sommando tutti i mintermini si coprono tutti gli 1 della funzione originaria e quello che si ottiene è una formula che ha la stessa tabella di verità della funzione originaria.

Es:

AB	A+B	min-termini
00	0	
01	1	$\sim AB = m1$
10	1	$A\sim B = m2$
11	1	$AB = m3$

$A+B = \sim AB + A\sim B + AB$  forma SOP

la forma POS si ottiene moltiplicando (AND) tutti i Maxtermini della funzione originaria. Maxtermini sono funzioni che vanno a 0 solo in corrispondenza di un solo 0 della funzione originaria. Facendo il prodotto di tutti i Maxtermini si ottiene una formula che ha la stessa tabella di verità della funzione originaria e quindi della forma SOP.

Es:

AB	A+B	Max-termini
00	0	$A+B=M0$
01	1	
10	1	
11	1	

$A+B = A+B$  forma POS

Negare la forma SOP significa considerare la funzione negata di quella originaria.  
Es.

AB	Not(A+B)	min-termini	Max-termini
00	1	$\sim A \sim B = m0$	
01	0		$A + \sim B = M1$
10	0		$\sim A + B = M2$
11	0		$\sim A + \sim B = M3$

$$\sim(SOP) = \sim(\sim AB + A \sim B + AB) = (A + \sim B)(\sim A + B)(\sim A + \sim B) = \sim(A+B) = \sim(POS)$$

che è la negazione della funzione originaria.

**In generale, ma non è l'unico modo, conviene sviluppare la POS applicando la distributiva una parentesi alla volta, semplificando dove possibile ad ogni passaggio, fino ad arrivare ad una somma di prodotti. Infine con assorbimento o elemento neutro si ricrea la forma SOP.**

Es.:

$$\begin{aligned} POS &= A+B = A1+B1 \\ &= A(\sim B+B)+B(A+\sim A) \\ &= A\sim B+AB+BA+B\sim A \\ &= A\sim B+\sim AB+AB = SOP \end{aligned}$$

Circa l'esempio da lei proposto,  $ABC+A\sim B\sim C$ , questa è una funzione che ha due 1 in corrispondenza di 111 e 100 mentre tutte le altre configurazioni sono a 0 quindi la SOP è  $m7+m4$  mentre la POS è  $M0 \cdot M1 \cdot M2 \cdot M3 \cdot M5 \cdot M6$  cioè'

$$(A+B+C)(A+B+\sim C)(A+\sim B+C)(\sim A+B+\sim C)(\sim A+\sim B+C)$$

Semplificando la SOP si ottiene:

$$A(BC+\sim B\sim C) = A\sim(B \text{ xor } C)$$

se vogliamo passare da POS a SOP:

POS

$$\begin{aligned} &(AA+AB+AC+BA+BB+BC+\sim CA+\sim CB+\sim CC)(A+\sim B+C)(A+\sim B+\sim C)(\sim A+B+\sim C)(\sim A+\sim B+C) \\ &(A+AB+AC+B+BC+A\sim C+B\sim C)(A+\sim B+C)(A+\sim B+\sim C)(\sim A+B+\sim C)(\sim A+\sim B+C) \\ &(A+B)(A+\sim B+C)(A+\sim B+\sim C)(\sim A+B+\sim C)(\sim A+\sim B+C) \\ &(AA+BA+A\sim B+B\sim B+AC+BC)(A+\sim B+\sim C)(\sim A+B+\sim C)(\sim A+\sim B+C) \\ &(A+BC)(A+\sim B+\sim C)(\sim A+B+\sim C)(\sim A+\sim B+C) \\ &(AA+ABC+A\sim B+BC\sim B+A\sim C+BC\sim C)(\sim A+B+\sim C)(\sim A+\sim B+C) \\ &(A)(\sim A+B+\sim C)(\sim A+\sim B+C) \\ &(A\sim A+AB+A\sim C)(\sim A+\sim B+C) \\ &(AB+A\sim C)(\sim A+\sim B+C) \\ &(AB\sim A+A\sim C\sim A+AB\sim B+A\sim C\sim B+ABC+A\sim CC) \\ &(A\sim C\sim B+ABC) \\ &ABC+A\sim B\sim C \end{aligned}$$

SOP

**Q:** Quando parliamo di una ALU che deve effettuare una sottrazione  $A-B$  ed in particolare del modulo ALU0 di una ALU a 32bit abbiamo una XOR che effettua la selezione di  $B$  oppure  $\sim B$  in base al segnale di controllo InvertiB. E' però necessario anche sommare 1 per avere il complemento a 2 di  $B$ . Mi pare di aver capito dalle slide [...] e dal testo che quando InvertiB è 1 viene settato ad 1 anche Rin per realizzare il complemento a 2. Ma questo non può essere fatto mettendo in OR Rin e InvertiB in modo da ottenere Rin quando InvertiB è 0 e 1 quando InvertiB è 1?

**A:** per realizzare il complemento a 2 di un numero  $b$  è necessario calcolare la formula  $\sim b + 1$ . Questo può essere fatto settando opportunamente il segnale Rin del FA che somma i bit 0. Se usiamo Rin per questo scopo non possiamo usarlo per riportare un eventuale riporto da un sommatore precedente, almeno non nel caso di sottrazioni.

Mettere in OR un segnale Rin in ingresso alla ALU e il segnale InvertiB significa implementare una tabella di questo tipo:

InvertiB	Risultato
0	$A+B+Rin$
1	$A-B$

In pratica con la sua soluzione quando InvertiB è a 1, l'eventuale Rin in ingresso verrebbe ignorato o, se preferisce, coperto dal segnale necessario per eseguire il complemento a 2.

**Q:** Ho cominciato lo sviluppo del progetto su Gatesim ma continuano a verificarsi dei crash e comportamenti imprevedibili da parte del simulatore. Sono convinto che non si tratti di un uso scorretto o di un'implementazione sbagliata. Ho anche fatto un'installazione di Windows XP su una macchina virtuale per capire se era un problema causato da Win7 (su Win7 ogni volta che il circuito diventava un po' più complesso non riuscivo più ad aprire il file e dovevo ricominciare dallo zero). Ora il problema nell'aprire il file non si presenta più, ma il simulatore richiede un tempo gigantesco per propagare i segnali (sto lavorando con clock vicini ai 15000ms per riuscire a far commutare un circuito particolarmente semplice). Comunque i crash continuano ad apparire ogni tanto. Date le difficoltà, vorrei chiedere se posso utilizzare un software simile, [...]

**A:**

<http://marchi.ricerca.di.unimi.it/Teaching/Architetture/>

...

Esercitazione: Introduzione a Gatesim & Circuiti combinatori

....

"Se avete problemi con la versione originale provate questa versione patchata di Gatesim"

Estratto dalle lezioni in laboratorio:

"Gatesim e' stato sviluppato usando standard americani. Risultato: può capitare che su grandi circuiti, specie se si utilizzano chip, le posizioni dei componenti vengano salvate nell'XML usando la virgola come separatore dei decimali e lette "all'americana" interpretando la virgola come separatore delle migliaia, trasformando es. 3,43561516666 in 343561516666. La patch che trovate sul sito risolve questo problema"

e poi...

"Cose da NON fare:

1) Usare i clock: Gatesim ha grossi problemi a gestire i clock, se vi occorre un clock sostituitelo con un pulsante da manovrare a mano.

....

5) scrivere una mail del tipo: "non mi funziona Gatesim". E' corretto invece scrivere "non mi funziona Gatesim, ho provato anche la patch ma il problema sussiste"

Detto questo, sarebbe utile se allegasse il suo circuito, così da avere un'idea di quale sia la criticità.