



Università degli studi di Parma
Dipartimento di Ingegneria dell'Informazione

Reti Logiche A a.a. 2007/2008

Algebra di Commutazione (ripasso)

prof. Stefano CASELLI

stefano.caselli@unipr.it

<http://www.ce.unipr.it/people/caselli>

<https://corsi.unipr.it>

Introduzione

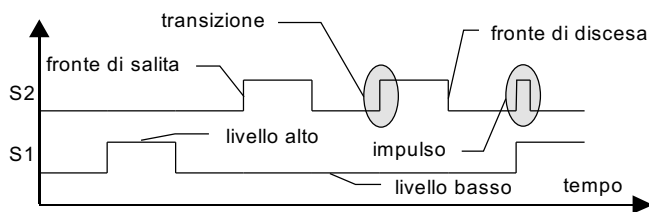


- Sistemi digitali
 - ▶ ottima immunità ai disturbi
 - ▶ facilità realizzativa
 - ▶ possibilità di creare metodologie di progetto automatizzabili
 - ▶ precisione prevedibile e arbitraria
- Tipi di sistemi
 - ▶ custom (antifurto, accensione auto, ...)
 - ▶ specializzati ma di uso generale (aritmetici, decoder, MUX)
 - ▶ con memoria (macchine a stati finiti, FSM)
 - ▶ senza memoria (circuiti combinatori)

Segnali binari



- Rappresentazione fisica (esempi)
 - ▶ tensione elettrica V
 - ▶ intensità di corrente I
 - ▶ potenza ottica P
- Diagramma temporale
 - ▶ trascureremo (quasi) sempre i transitori



Algebra di Commutazione

- 3 -

Algebra di Boole e algebra di commutazione



- Algebra di Boole
 - ▶ insieme di elementi K
 - ▶ esistono due funzioni $\{+, \bullet\}$ che fanno corrispondere a una qualsiasi coppia di elementi di K un elemento di K
 - ▶ Una funzione $\{\bar{}\}$
- Algebra di commutazione
 - ▶ I valori delle variabili di commutazioni possono assumere solo due valori $(0,1)$, (V,F) , (H,L) , ...
 - ▶ la variabile logica è un numero binario e gode di diverse proprietà
 - ▶ esiste una corrispondenza fra gli operatori fondamentali dell'algebra di commutazione e i circuiti digitali

Algebra di Commutazione

- 4 -

Assiomi dell'algebra di Boole (1)



- K contiene almeno due elementi a e b tali che $a \neq b$
- Chiusura
 - ▶ per ogni a e b in K : $a+b \in K$ e $a \cdot b \in K$
- Proprietà commutativa
 - ▶ $a + b = b + a$ e $a \cdot b = b \cdot a$
- Proprietà associativa
 - ▶ $(a + b) + c = a + (b+c) = a + b + c$
 - ▶ $a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$

Assiomi dell'algebra di Boole (2)



- Elementi identità
 - ▶ Esiste un elemento identità rispetto a $\{+\}$, 0 , tale che $a + 0 = a$ per ogni $a \in K$
 - ▶ Esiste un elemento identità rispetto a $\{\cdot\}$, 1 , tale che $a \cdot 1 = a$ per ogni $a \in K$
- Proprietà distributiva
 - ▶ $a + (b \cdot c) = (a + b) \cdot (a + c)$
 - ▶ $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
- Complemento

Per ogni $a \in K$ esiste un elemento $a' \in K$ tale che:

$$\text{▶ } (a + a') = 1 \quad \text{e} \quad (a \cdot a') = 0$$

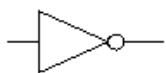
Algebra di commutazione



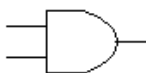
- L'insieme K è ristretto a solo due elementi $K=\{0, 1\}$
- Le operazioni logiche fondamentali OR, AND, NOT soddisfano gli assiomi dell'algebra di Boole
- Porte logiche: elementi circuitali corrispondenti

A	$f(A)=\overline{A}$	A	B	$f(A,B)=A \cdot B$	A	B	$f(A,B)=A+B$
1	0	0	0	0	0	0	0
0	1	0	1	0	0	1	1
		1	0	0	1	0	1
		1	1	1	1	1	1

NOT
negazione



AND
prodotto logico



OR
somma logica



Algebra di Commutazione

- 7 -

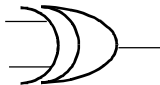
Altri operatori di uso comune



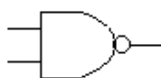
- Esistono 16 funzioni di due variabili, corrispondenti alle combinazioni dei due ingressi
- Le più interessanti sono: XOR, NAND, NOR, XNOR

A	B	$f(A,B)=A \oplus B$	A	B	$f(A,B)=\overline{A \cdot B}$	A	B	$f(A,B)=\overline{A+B}$
0	0	0	0	0	1	0	0	1
0	1	1	0	1	1	0	1	0
1	0	1	1	0	1	1	0	0
1	1	0	1	1	0	1	1	0

EX-OR



NAND



NOR



Algebra di Commutazione

- 8 -

Funzioni di due variabili



- Esercizio:
- Le possibili funzioni di n variabili sono 2^{2^n}
 - ▶ perchè?
 - ▶ enumerare tutte le funzioni di due variabili

Algebra di commutazione: proprietà



- La dimostrazione può avvenire
 - ▶ mediante analisi esaustiva
 - ▶ usando proprietà già definite
- Principio di dualità
 - ▶ se vale un'identità booleana, allora vale anche l'identità duale, ottenuta scambiando $+$ con $\cdot$ (somma con prodotto) e la costante 0 con la costante 1:
 $0 \leftrightarrow 1$ e $+$ $\leftrightarrow$ $\cdot$
 - ▶ è conseguenza dell'interscambiabilità degli assiomi dell'algebra di Boole

Algebra di commutazione: Riepilogo proprietà



N.	Descrizione	Nome
1	Esistono gli elementi $0, 1 \in K$ tali che : $A + 0 = A$ $A \cdot 1 = A$	Esistenza elementi identità
2	$A + B = B + A$ $A \cdot B = B \cdot A$	Proprietà commutativa
3	$A \cdot (B + C) = A \cdot B + A \cdot C$ $A + (B \cdot C) = (A + B) \cdot (A + C)$	Proprietà distributiva
4	Per ogni $A \in K$, esiste $\bar{A}$ tale che: $A \cdot \bar{A} = 0$ e $A + \bar{A} = 1$	Esistenza dell'inverso

Algebra di commutazione: Riepilogo proprietà



N.	Descrizione	Nome
5	$A + (B + C) = (A + B) + C$ $A (BC) = (AB) C$	Proprietà associativa
6	$A + A = A$ $A \cdot A = A$	Proprietà dell'idempotenza
7	$\overline{A + B} = \bar{A} \cdot \bar{B}$ $\overline{A \cdot B} = \bar{A} + \bar{B}$	Legge di DeMorgan
8	$\overline{\bar{A}} = A$	Involuzione

Funzioni logiche vs porte logiche



- Funzione logica a singola uscita $z = f(x_1, x_2, \dots, x_n)$
 - ▶ Legge che associa un valore binario a tutte le combinazioni delle variabili indipendenti
 - ▶ Astrazione che non considera la "dinamica" dei segnali
- Qualunque funzione logica può realizzarsi usando un *insieme completo* di operatori elementari
 - ▶ NAND, NOR, (AND, NOT), (OR, NOT), (AND, OR, NOT)
 - ▶ Combinazioni di porte logiche consentono di realizzare le funzioni logiche
- Vedremo anche come trattare i casi con ingressi non completamente specificati e uscite multiple

Esempio: rilevatore di maggioranza



- Progettare un circuito logico a 3 ingressi (A, B, C) e una uscita U che assuma valore 1 quando, all'ingresso, il numero degli 1 supera il numero degli 0
- L'attenzione è sugli 1 della tabella di verità

Tabella di verità

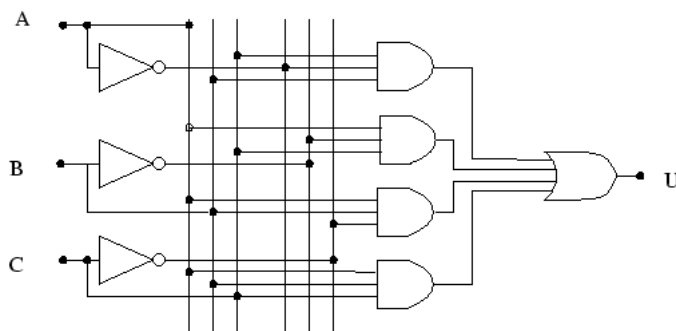
A	B	C	U
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$U = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$\bar{A}BC$
 $A\bar{B}C$
 $AB\bar{C}$
 ABC

$\nearrow$
 $\nearrow$
 $\rightarrow$
 $\rightarrow$

Rilevatore di maggioranza: Rappresentazione circuitale



$$U = \overline{A}BC + \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C}$$

Rilevatore di magg.: soluzione duale (1)



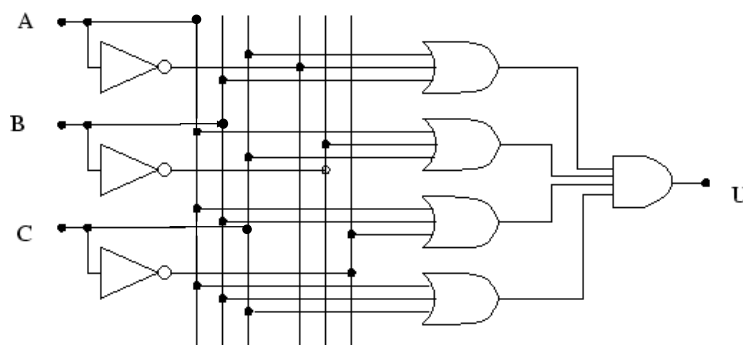
- L'attenzione è sugli 0 della tabella di verità

A	B	C	U
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$\begin{aligned}
 &A + B + C \\
 &\xrightarrow{\quad\quad\quad} A + B + \overline{C} \\
 &\xrightarrow{\quad\quad\quad} A + \overline{B} + C \\
 &\xrightarrow{\quad\quad\quad} \overline{A} + B + C
 \end{aligned}$$

$$U = (\overline{A} + B + C) \cdot (A + \overline{B} + C) \cdot (A + B + \overline{C}) \cdot (A + B + C)$$

Rilevatore di magg.: soluzione duale (2)



$$U = (\bar{A} + B + C) \cdot (A + \bar{B} + C) \cdot (A + B + \bar{C}) \cdot (A + B + C)$$

Reti Combinatorie: definizione generale



- Circuito privo di retroazioni, formato collegando circuiti combinatori elementari
- Le porte logiche elementari (ad es. quelle booleane OR, AND e NOT) sono circuiti combinatori
- Se $m = 1$ la rete combinatoria si dice "a uscita *singola*", altrimenti si dice "a uscite *multiple*"



Analisi di una rete combinatoria



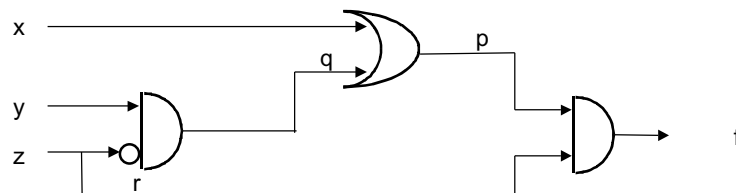
- *Scomposizione:*
 - ▶ ogni RC ad n ingressi ed m uscite può essere scomposta nel parallelo di m RC, ciascuna ad n ingressi
- *Simulazione del funzionamento:*
 - ▶ si assegnano valori agli ingressi della rete propagandoli in avanti, fino a determinare il valore logico dell'uscita
- *Valutazione:*
 - ▶ si ripete l'operazione per tutte le 2^n configurazioni delle variabili di ingresso

Equivalenza fra espressioni booleane e reti combinatorie



- A ogni RC($x_1, x_2, \dots, x_n$) a *una* uscita e a n ingressi $x_1, x_2, \dots, x_n$, si può sempre associare una e una sola espressione booleana EB($x_1, x_2, \dots, x_n$) a n variabili $x_1, x_2, \dots, x_n$, tale che
 - ▶ $\forall$ assegnamento A agli ingressi $x_1, x_2, \dots, x_n$ della rete combinatoria RC si ha:
 $RC(A) = EB|_A$ e viceversa
 - ▶ EB contiene operatori logici, variabili e costanti con lo stesso flusso di elaborazione, eventualmente precisato attraverso parentesi, presente nella RC e viceversa

Costruzione dell'EB a partire dalla RC



$$\begin{cases} p = x+q \\ q = y \cdot r \\ r = \bar{z} \\ f = p \cdot z \end{cases} \quad \begin{cases} p = x+q \\ q = y \cdot \bar{z} \\ r = \bar{z} \\ f = p \cdot z \end{cases} \quad \begin{cases} p = x+y \cdot \bar{z} \\ q = y \cdot \bar{z} \\ r = \bar{z} \\ f = p \cdot z \end{cases} \quad \begin{cases} p = x+y \cdot \bar{z} \\ q = y \cdot \bar{z} \\ r = \bar{z} \\ f = x+y \cdot \bar{z} \cdot z \end{cases}$$

$$f = (x + y \cdot \bar{z}) \cdot z$$

Funzioni, espressioni, schemi logici



- Una funzione logica è definita univocamente dalla sua tabella della verità
- Ad ogni funzione possono corrispondere infinite espressioni logiche
- Ad ogni espressione corrisponde una ed una sola funzione logica completamente specificata (ottenibile mediante valutazione)
- Espressioni corrispondenti alla medesima funzione sono tra loro equivalenti
- Ad ogni espressione corrisponde uno ed uno solo schema logico e viceversa

Livelli di una RC



- Espressione a due livelli
 - ▶ contiene solo due livelli di operatori annidati (trascurando la negazione)
- Espressione a più livelli
 - ▶ contiene più di due livelli di operatori annidati (trascurando la negazione)
- Esempi (2 e 3 liv.): $f = x + y \bullet \bar{z}$ $f = x + y \bullet (\bar{z} + u)$
- Il numero di livelli influenza
 - ▶ costo realizzativo
 - ▶ velocità circuito

Equivalenze fra espressioni booleane



- Due espressioni booleane $f(x_1, x_2, \dots, x_n)$ e $g(x_1, x_2, \dots, x_n)$, a $n \geq 1$ variabili, sono *equivalenti* se e solo se ammettono la stessa tabella delle verità

- Esempio

x	y	z	f	g
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	1
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

$$f(x, y, z) = x + y \cdot z$$

$$g(x, y, z) = x + \bar{x} \cdot y \cdot z$$

Le due RC sono *funzionalmente* equivalenti ma sono *differenti*, per es. in termini di costo

Esempio di criterio di scelta: #letterali



- Criterio di costo (dei letterali) di una rete combinatoria a due livelli
 - ▶ costo = # degli ingressi nel primo livello della rete
 - ▶ vale solo per funzioni booleane a 2 livelli
- Data una funzione booleana, esistono più (infinite) reti combinatorie che la realizzano
- Problema
 - ▶ sintetizzare la rete comb. di costo minimo
- Esempio: $f(x, y, z) = x + y \cdot z$ $\text{costo}(f) = 1 + 2 = 3$
 $g(x, y, z) = x + \bar{x} \cdot y \cdot z$ $\text{costo}(g) = 1 + 3 = 4$

Un criterio alternativo: # porte logiche, # morsetti di ingresso



- E' correlato al criterio #letterali
- E' correlato agli aspetti fisici di area e wiring
- E' il criterio ottimizzato dagli algoritmi classici di sintesi
- Il numero di porte logiche o blocchi N_B prevale sul numero di morsetti di ingresso N_m
- Ad es.: $f(x, y, z) = x + y \cdot z$, $g(x, y, z) = x + x' \cdot y \cdot z$
 - ▶ Costo f: $N_B = 2$, $N_m = 4$
 - ▶ Costo g: $N_B = 2$, $N_m = 5$
 - ▶ f "costa meno" di g in base al criterio N_m

Un criterio alternativo: # porte logiche, # morsetti di ingresso



- $f(x, y, z, w) = x \cdot w' + y \cdot z$
 $g(x, y, z, w) = x + w' + x' \cdot y \cdot z \cdot w$
 - ▶ Costo f: $N_B = 3$, $N_m = 6$
 - ▶ Costo g: $N_B = 2$, $N_m = 7$
- g "costa meno" di f in base al criterio N_B , che prevale su N_m