

Componenti per l'elaborazione binaria dell'informazione

Approfondimento del corso di reti logiche

M. Favalli

Engineering Department in Ferrara



Navigation icons: back, forward, search, etc.

M. Favalli (ENDIF)

Gate

Analisi e sintesi dei circuiti digitali

1 / 41

Porte logiche

Sommario

- 1 Porte logiche
- 2 Il livello switch
- 3 Aspetti tecnologici
- 4 Reti logiche combinatorie
 - Analisi
 - Sintesi

Navigation icons: back, forward, search, etc.

M. Favalli (ENDIF)

Gate

Analisi e sintesi dei circuiti digitali

3 / 41

Sommario

- 1 Porte logiche
- 2 Il livello switch
- 3 Aspetti tecnologici
- 4 Reti logiche combinatorie
 - Analisi
 - Sintesi

Navigation icons: back, forward, search, etc.

M. Favalli (ENDIF)

Gate

Analisi e sintesi dei circuiti digitali

2 / 41

Porte logiche

Porte logiche

Si può dare una definizione duplice delle porte logiche (gate):

Visione indipendente dalla tecnologia

Semplici funzioni dell'algebra di commutazione con le quali costruire reti logiche che realizzano funzioni più complesse

Visione dipendente dalla tecnologia

Circuiti elementari di tipo digitale messi a disposizione da una data tecnologia

Navigation icons: back, forward, search, etc.

M. Favalli (ENDIF)

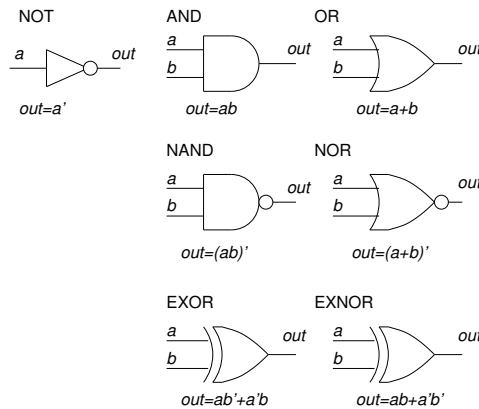
Gate

Analisi e sintesi dei circuiti digitali

4 / 41

Porte logiche

numero di ingressi (fan-in) = 2



Navigation icons: back, forward, search, etc.

Aspetti tecnologici

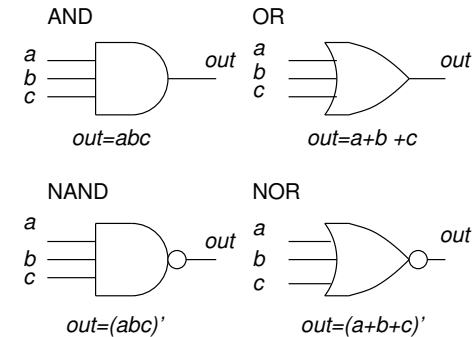
- Esistono diverse tecnologie microelettroniche che consentono di realizzare gate (TTL, ECL, MOS, CMOS)
- Ciascuna tecnologia consente di realizzare in maniera efficiente solo un certo numero di tipi di gate (gli altri sono realizzabili mediante opportune reti di gate elementari)
- Al livello tecnologico i gate non sono caratterizzati solo dalle funzioni che realizzano, ma anche da altri parametri (costo, ritardo, consumo di potenza)
- Per capire questi aspetti analizzeremo una tecnologia in dettaglio

Navigation icons: back, forward, search, etc.

Porte logiche

numero di ingressi (fan-in) > 2

Le porte logiche possono essere estese al caso in cui siano presenti più ingressi



Navigation icons: back, forward, search, etc.

Sommario

- 1 Porte logiche
- 2 Il livello switch
- 3 Aspetti tecnologici
- 4 Reti logiche combinatorie
 - Analisi
 - Sintesi

Navigation icons: back, forward, search, etc.

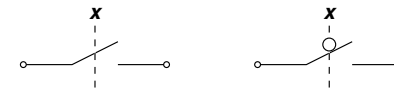
Livello switch

- La tecnologia elettronica più diffusa è attualmente quella CMOS
- In tale tecnologia il componente elementare più semplice descrivibile al livello logico non è il gate, ma il transistor descrivibile come un interruttore (switch)
- Questo ci consente di vedere il modo in cui una porta logica è costruita

Livello switch

- Supponiamo che un segnale binario (x) possa controllare lo stato (ON, OFF) dell'interruttore
- Nella tecnologia CMOS sono presenti due tipi di transistori

tipo n		tipo p	
x	stato	x	stato
0	OFF	0	ON
1	ON	1	OFF



Descrizione di reti di switch mediante l'algebra di commutazione

- I sistemi fisici costituiti da switch interconnessi sono descrivibili utilizzando l'algebra di commutazione
- Bisogna codificare gli stati OFF e ON con un valore binario

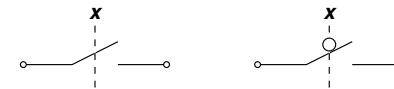
logica positiva		logica negativa	
0	OFF	0	ON
1	ON	1	OFF

- Per descrizione della rete, si intende un'espressione che assume il valore 1 se la rete è ON (logica positiva)

Livello switch

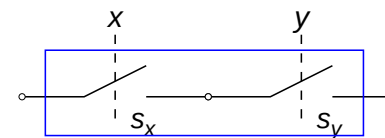
- Supponiamo che un segnale binario (x) possa controllare lo stato (ON, OFF) dell'interruttore
- Nella tecnologia CMOS sono presenti due tipi di transistori

tipo n		tipo p	
x	stato	x	stato
0	OFF	0	ON
1	ON	1	OFF



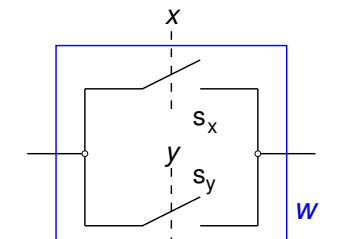
Connessioni serie e parallelo

Connessione serie:



s_x	s_y	serie
OFF	OFF	OFF
OFF	ON	OFF
ON	OFF	OFF
ON	ON	ON

Connessione parallelo:



s_x	s_y	parallelo
OFF	OFF	OFF
OFF	ON	ON
ON	OFF	ON
ON	ON	ON

Rappresentazione logica

Logica positiva	s_x	s_y	serie	s_x	s_y	parallelo
	0	0	0	0	0	0
	0	1	0	0	1	1
	1	0	0	1	0	1
	1	1	1	1	1	1

Logica negativa	AND			OR		
	s_x	s_y	serie	s_x	s_y	parallelo
	1	1	1	1	1	1
	1	0	1	1	0	0
	0	1	1	0	1	0
	0	0	0	0	0	0

OR			AND		
s_x	s_y	serie	s_x	s_y	parallelo
1	1	1	1	1	1
1	0	1	1	0	0
0	1	1	0	1	0
0	0	0	0	0	0

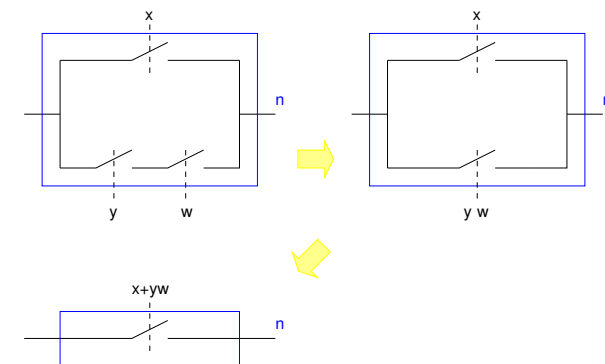
Principio di dualità

- Utilizzando lo stesso tipo di logica, le reti serie e parallelo realizzano funzioni duali (AND, OR)
- Cambiando tipo di logica, la stessa rete viene descritta da una funzione duale
- Il principio di dualità riguarda il modello logico dei sistemi fisici
- Punto di vista del calcolo delle proposizioni
 - La serie é ON se entrambi gli switch sono ON.
 - Il parallelo é ON se uno o l'altro degli switch é ON.
 - La serie é OFF se uno degli switch é OFF.
 - Il parallelo é OFF se entrambi gli switch sono OFF.

Reti di switch

- Reti di switch in serie e in parallelo possono essere composte in reti piú complesse
- Tali reti possono essere analizzate sostituendo iterativamente:
 - 1 a ciascuna serie di due o piú switch un singolo switch controllato dal prodotto logico degli ingressi degli switch sostituiti
 - 2 a ciascun parallelo di due o piú switch un singolo switch controllato dalla somma logica degli ingressi degli switch sostituiti

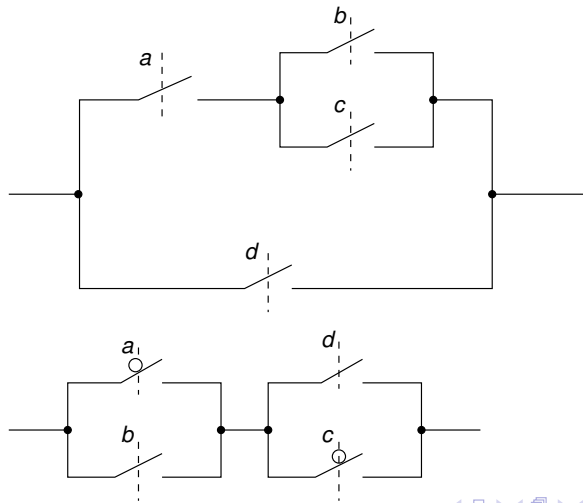
Reti di switch (esempio di analisi)



Si può quindi affermare che la rete n é ON se é vera $x + yw$

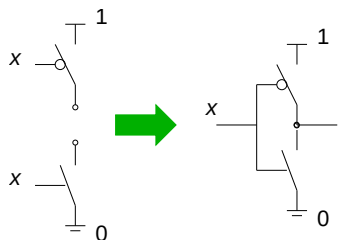
Esercizi di analisi

Si analizzino queste reti di switch nell'ipotesi di utilizzare una logica positiva



Gate (tecnologia CMOS)

- La soluzione consiste nel fare in modo che una rete di switch possa controllare gli switch di un'altra rete (con un guadagno in grado di "rinforzare" i segnali)
- Vediamo come esempio il caso della tecnologia CMOS
- Procuriamoci le costanti 1 (vdd) e 0 (gnd) e utilizziamo switch di tipo *p* e *n* per realizzare due reti con stati complementari



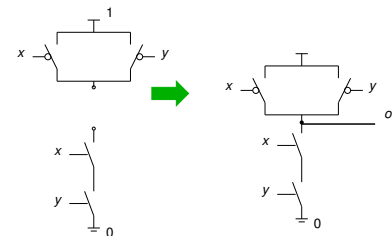
x	pull-down (n)	pull-up (p)
0	OFF	ON
1	ON	OFF

Problema

- Le reti di switch sono piuttosto interessanti, ma c'è un problema di carattere elettrico.
- Gli switch non sono ideali ma hanno una resistenza serie per cui l'informazione si degrada attraversando serie di con troppi switch
- Come realizzare espressioni complesse?
- Si farà riferimento alla logica positiva

NAND

- Nella tecnologia CMOS non si può realizzare direttamente un gate AND
- La realizzazione di un NAND è simile a quella di un NOT: si costruiscono due reti complementari
 - pull-up: che pilota un 1 uno degli ingressi vale 0
 - pull-down: che pilota uno 0 quando entrambi gli ingressi valgono 1



xy	pull-down (n)	pull-up (p)
00	OFF	ON
01	OFF	ON
10	OFF	ON
11	ON	OFF

Esercizi

Si realizzino porte logiche CMOS per le seguenti funzioni logiche:

- $out = (x + y)'$
- $out = (x \cdot (w + y))'$
- $out = a + b$

Sommario

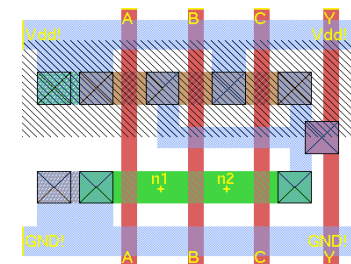
- 1 Porte logiche
- 2 Il livello switch
- 3 Aspetti tecnologici
- 4 Reti logiche combinatorie
 - Analisi
 - Sintesi

Circuiti integrati

- Alcune informazioni sulle porte logiche che possono essere dedotte considerando la loro struttura al livello switch
- Una ulteriore informazione riguarda la tecnologia:
 - i circuiti sono realizzati integrando numerose porte logiche ($> 10^6$) in un circuito integrato realizzato con materiali semiconduttori (es. silicio)
 - la tecnologia é planare: i dispositivi (switch, gate) non possono essere sovrapposti (le interconnessioni invece possono occupare diversi livelli)
 - il costo é proporzionale all'area utilizzata che dipende dal numero di dispositivi e dalle interconnessioni

Costo

- Nella tecnologia CMOS un gate corrisponde fisicamente a un area rettangolare le cui dimensioni sono proporzionali al numero di transistori e quindi di ingressi
- In una rete complessa, un primo contributo al costo é dato da un termine proporzionale alla somma dei gate pesata sui loro ingressi
- Si vedrá in seguito il contributo delle interconnessioni



Esempio di NAND a 3 ingressi

Analisi e sintesi di reti combinatorie

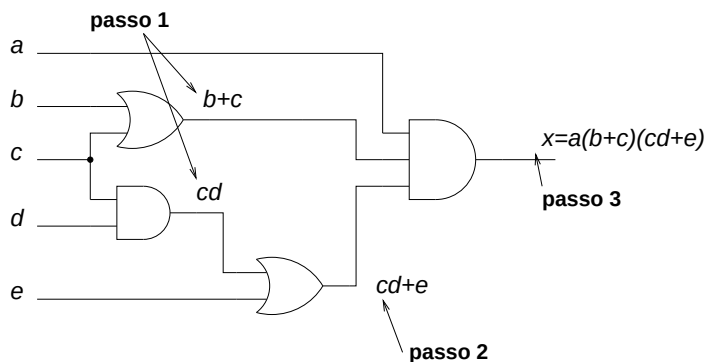
- Per analisi di una rete combinatoria si intende il processo con il quale partendo dalla rete si ottiene la funzione corrispondente
- Per sintesi di una rete combinatoria si intende il processo con il quale partendo da una funzione e da eventuali obiettivi di progetto (costo, ritardo) si ottiene una rete
- In entrambi i casi, le espressioni dell'algebra di commutazione sono il modello matematico che consente di gestire le trasformazioni che si hanno nei due processi

Algoritmo di analisi

Rete logica $\Rightarrow$ Espressione $\Rightarrow$ Funzione
algoritmo valutazione

- 1 Si assegna un nome a ciascun ingresso
- 2 Si procede in maniera iterativa dagli ingressi del circuito esprimendo l'uscita di ciascun gate come un'espressione degli ingressi della rete
 - a ciascun passo vengono considerati tutti i gate i cui ingressi sono già stati calcolati
- 3 Si procede fino a quando non si ha l'espressione di tutte le uscite della rete

Esempio di analisi



Algoritmo di sintesi - I

Funzione $\Rightarrow$ Espressione $\Rightarrow$ Rete logica
algoritmo

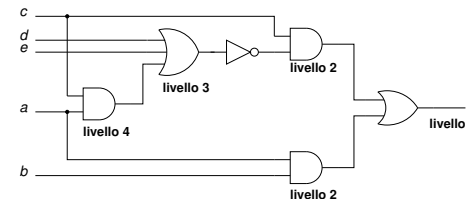
- 1 Si inseriscono nell'espressione tutte le parentesi comprese quelle rese non necessarie dalle regole dell'algebra di commutazione (senza considerare le complementazioni)
- 2 Si analizza l'espressione determinando il livello di ciascun operatore binario
 - 1 si inizializza un indice di livello a 1 o a 0 (se tutta l'espressione è racchiusa fra parentesi)
 - 2 si incrementa di 1 tale indice tutte le volte che si incontra una parentesi aperta e lo si decrementa di 1 tutte le volte che si incontra una parentesi chiusa

Algoritmo di sintesi - II

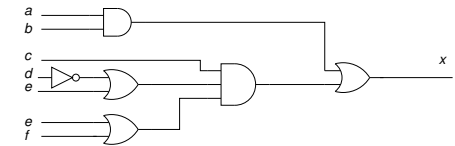
- 3 Si disegnano i segnali di ingresso corrispondenti alle diverse variabili dell'espressione
- 4 Partendo dal valore piú alto dell'indice di livello, si disegnano i simboli dei gate corrispondenti agli operatori connettendo gli opportuni segnali di ingresso a tale gate
- 5 Nel caso una variabile di ingresso o una parentesi risultino complementate si aggiunge un invertitore

Esempi di sintesi

$$\begin{aligned}
 x &= ab + c(d + e + ac)' \\
 &= (a \cdot b) + (c \cdot (d + e + (a \cdot c)))' \\
 &= {}_1(a \cdot {}_2b) + {}_1(c \cdot {}_2(d + {}_3e + {}_3(a \cdot {}_4c)))'
 \end{aligned}$$



$$\begin{aligned}
 x &= ab + c(d' + e)(f + g) \\
 &= (a \cdot b) + (c \cdot (d' + e) \cdot (f + g)) \\
 &= {}_1(a \cdot {}_2b) + {}_1(c \cdot {}_2(d' + {}_3e) \cdot {}_2(f + {}_3g))
 \end{aligned}$$



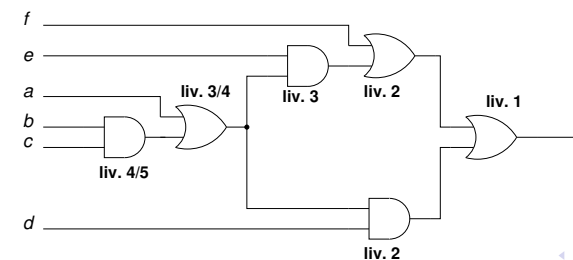
Il ruolo del fan-out

- É possibile che una porta logica ne piloti piú di una (fan-out > 1)
- Questo consente di ridurre il numero di porte logiche e di conseguenza il costo di una rete
- L'unico problema é un aumento del ritardo della rete
- Dal punto di vista degli algoritmi di analisi non cambia nulla
- Dal punto di vista della sintesi, invece, si hanno dei cambiamenti
- In particolare, prima del passo 3, si inserisce un ulteriore passo di ricerca di sottoespressioni comuni (si noti che l'indice di livello puó essere diverso)
- Per il momento, questa operazione verrá svolta su basi intuitive

Sintesi in presenza di fan-out

$$\begin{aligned}
 x &= (a + bc)d + ((a + bc)e + f) \\
 &= ((a + (bc))d) + (((a + (bc))e) + f) \\
 &= ({}_1(a + {}_3(b \cdot {}_4c)) \cdot {}_2d) + {}_1(({}_1(a + {}_4(b \cdot {}_5c)) \cdot {}_3e) + {}_2f)
 \end{aligned}$$

Nel disegnare la rete conviene considerare la sottoespressione con i livelli maggiori



Livelli

Metodo per il calcolo dei livelli

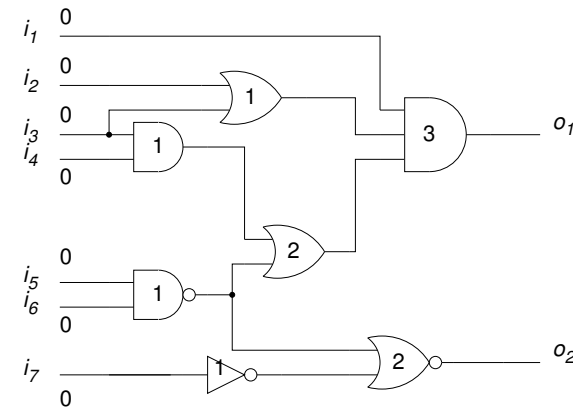
Una rete può essere attraversata dagli ingressi verso le uscite o viceversa (metodo precedente) assegnando a ciascun gate un indice di livello

Entrambi i tipi di indice sono utili in diverse operazioni eseguibili sulle reti.

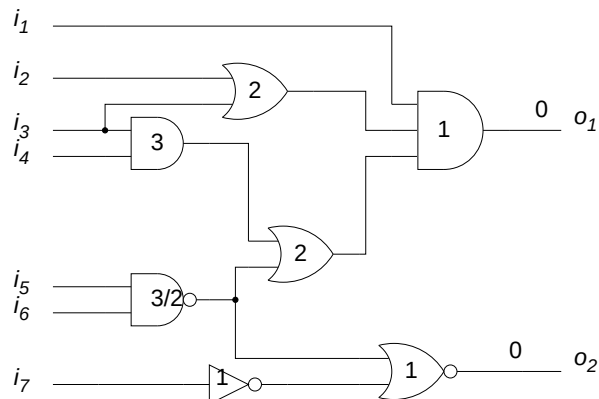
Nel caso di attraversamento dagli ingressi (livello 0) verso le uscite, il livello di ciascun gate è definito univocamente

Nel caso di attraversamento dalle uscite (livello 0) verso gli ingressi il livello non è definito univocamente

Livelli: dagli ingressi verso le uscite



Livelli: dalle uscite verso gli ingressi



Esercizi

Conclusioni

- Si é visto come si possa passare da un espressione a una rete o da una rete a un espressione a una funzione (tramite la valutazione)
- Non sappiamo ancora come passare da una funzione a un espressione
- Questo passo é quello piú rilevante dal punto di vista della sintesi