Il Livello Logico Digitale (i)

A cura di:
Luca Breveglieri Giuseppe Pozzi

DEI, Politecnico di Milano
luca.breveglieri/giuseppe.pozzi@polimi.it
- versione del 7 aprile 2003 -

Aritmetica binaria

Sommario

- Aritmetica binaria:
  - conversioni di base di numero:
    - base 2;
    - base 8;
    - base 10;
    - base 16
  - rappresentazione in numero e segno;
  - rappresentazione in complemento a 1;
  - rappresentazione in complemento a 2.
- Vantaggi della rappresentazione in complemento a 2.

Conversione in base 2

- In base 10:
  - $75 = 7*10^1 + 5*10^0$
- In base 2:
  - $110 = 1*2^2+1*2^1+0*2^0=6$
- Conversione del numero 26 da base 10 base 2:
  - 26/2 = 13 resto 0
  - 13/2 = 6 resto 1
  - 6/2 = 3 resto 0
  - 3/2 = 1 resto 1
  - 1/2 = 0 resto 1
  - Il numero 26_{10} diventa 11010_2

Basi superiori a 10

- Nelle basi superiori a 10 il dizionario dei simboli viene definito aggiungendo alle cifre arabe 0 .. 9 i caratteri alfabetici A, B…
- Pertanto la numerazione in base 16 prevede i simboli 0, 1 .. 9, A, B, C, D, E, F.

Base 8 e base 16

- Dalla rappresentazione in base 2 si passa alla rappresentazione in base 8 raggruppando in blocchi di 3 i bit:
  - 11010_2 diventa: 011 010_2 ==>32_8 ;
- Dalla rappresentazione in base 2 si passa alla rappresentazione in base 16 raggruppando in blocchi di 4 i bit:
  - 11010_2 diventa: 0001 1010_2 ==>$1A_{16}$.

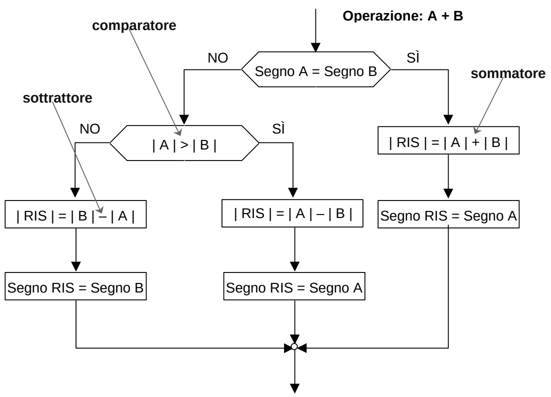
Modulo e segno

- Il primo bit indica il segno:
  - 0 per un numero positivo;
  - 1 per un numero negativo.
- I rimanenti N-1 bit rappresentano il modulo del numero.
- Es:
  - 0 110: numero positivo, $1*2^2+1*2^1+0*2^0=+6$
  - 1 110: numero negativo, $1*2^2+1*2^1+0*2^0=-6$

Modulo e segno

- Si hanno due rappresentazioni del numero 0:
  - 0000 equivalente a +0;
  - 1000 equivalente a -0.
- Con N bit rappresento i numeri inclusi nell'intervallo $-2^{N-1}-1$ .. $2^{N-1}-1$, estremi compresi.
- Le operazioni aritmetiche di somma e sottrazione richiedono un'analisi del segno del numero prima di poter procedere.

Modulo e segno



Complemento a 1

- Usa una codifica che intrinsecamente indica sia il numero sia il segno.
- È ancora ridondante perchè si ha doppia rappresentazione del numero 0.
- Se il numero è positivo, la sua rappresentazione in complemento a 1 concide con la rappresentazione in modulo e segno.

Complemento a 1

- Se il numero è negativo, la rappresentazione in complemento a 1 è calcolata complementando a 1 tutti i bit del modulo, invertendoli.
- Esempi:
  - il numero +9 (in modulo e segno è 01001) viene rappresentato in complemento a 1 come 01001;
  - il numero -9 (in modulo e segno è 11001) viene rappresentato in complemento a 1 come 10110.

Complemento a 1

- Per effettuare la somma tra due numeri A e B, procedo normalmente.
- Per effettuare la sottrazione, calcolo l’opposto del secondo e procedo, effettuando la somma A + (-B).
- Il circuito integrato richiesto è più semplice.

Rappresentazioni a confronto

Codifica	Modulo e segno	Complemento a 1
0000	+0	+0
0001	+1	+1
0010	+2	+2
0011	+3	+3
0100	+4	+4
0101	+5	+5
0110	+6	+6
0111	+7	+7
1000	-0	-7
1001	-1	-6
1010	-2	-5
1011	-3	-4
1100	-4	-3
1101	-5	-2
1110	-6	-1
1111	-7	-0

Osservazione

- Quando ricevo un numero in formato binario, prima di interpretarlo devo conoscere in quale rappresentazione viene fornito.

Complemento a 2

- Se il numero è positivo, la rappresentazione in complemento a 2 concide con la rappresentazione in modulo e segno.

Complemento a 2

- Se il numero è negativo, la rappresentazione in complemento a 2 è calcolata invertendo tutti i bit ed aggiungendo 1.
- Esempi:
  - il numero +9 (in modulo e segno è 01001) viene rappresentato in complemento a 2 come 01001;
  - il numero -9 (in modulo e segno è 11001) viene rappresentato in complemento a 2 come 10110+1 = 10111.

Complemento a 2

- La rappresentazione non è simmetrica e non è ridondante.
- Con N bit rappresento i numeri inclusi nell'intervallo -2^{N-1} .. $2^{N-1}-1$, estremi compresi.
- Effettuando la somma tra due numeri ottengo sempre il risultato corretto.

Osservazione

- Si ipotizza sempre che il risultato delle operazioni di somma e sottrazione possa essere memorizzato correttamente nei bit a disposizione.

Estensione del segno

- Per i numeri in complemento a 2 si estende il segno riportando il bit del segno tante volte quanti sono i bit da aggiungere.
- Es:
  - estendere a 8 bit il numero 01001:00001001
  - estendere a 8 bit il numero 10111:11110111

Rappresentazioni a confronto

Codifica	M e s	Compl a 1	Compl a 2
0000	+0	+0	+0
0001	+1	+1	+1
0010	+2	+2	+2
0011	+3	+3	+3
0100	+4	+4	+4
0101	+5	+5	+5
0110	+6	+6	+6
0111	+7	+7	+7
1000	-0	-7	-8
1001	-1	-6	-7
1010	-2	-5	-6
1011	-3	-4	-5
1100	-4	-3	-4
1101	-5	-2	-3
1110	-6	-1	-2
1111	-7	-0	-1

Complemento a 2

- Esempi:

  011001 +
  000011 =
  011100 OK: 25+3=28

  011001 +
  111101 =
  (1)010110 OK: 25-3=22
  trascuro il riporto (carry)

Complemento a 2

- Esempi:

  011001 +
  001000 =
  100001 Err: 25+8=33 e non -31

- Se la somma di due numeri positivi genera un numero negativo (o di due negativi genera un positivo) si ha trabocco (overflow): il risultato ottenuto è errato.

Moltiplicazione per 2

- Dato un numero naturale, lo si moltiplica per 2 aggiungendo uno 0 a destra ed effettuando uno scorrimento a sinistra dei bit.
- Es:
  - 01001 x 10 = 10010
- Attenzione al supero di capacità.

Divisione per 2

- Dato un numero naturale, lo si divide per 2 eliminando la cifra meno significativa, che costituisce il resto.
- Es:
  - 01001 : 10 = 0100, resto 1

Esercizio

- Verificare se il meccanismo di moltiplicazione per due e di divisione per due vale anche per i numeri in complemento a 2.

Porte logiche

Segnali e informazioni

- Per elaborare informazioni, occorre rappresentarle (o codificarle)
- Per rappresentare (o codificare) le informazioni si usano segnali
- I segnali devono essere elaborati, nei modi opportuni, tramite dispositivi di elaborazione

Il segnale binario

- Segnale binario: una grandezza che può assumere due valori distinti, convenzionalmente indicati con 0 e 1

  $s \in \{0, 1\}$

- Qualsiasi informazione è rappresentabile (o codificabile) tramite uno o più segnali binari (per esempio i caratteri del codice ASCII)

Il segnale binario

- Rappresentazione fisica del segnale binario: si usano svariate grandezze fisiche
  - tensione elettrica (la più usata!)
  - corrente elettrica
  - potenza ottica
  - e altre grandezze fisiche ancora ...

Il segnale binario

- Elaborazione del segnale binario: si usano svariate classi di dispositivi di elaborazione
  - porte logiche
  - reti combinatorie
  - reti sequenziali
- Sono tutti circuiti digitali (o numerici)

Circuiti digitali

- L'elaborazione di segnali (o informazioni) binarie è oggi svolta principalmente tramite tecnologie microelettroniche (e in parte anche ottiche)
- I circuiti microelettronici che elaborano segnali (o informazioni) binari si chiamano circuiti digitali (o circuiti numerici, o circuiti logici)

Porte logiche

- I circuiti digitali sono formati da componenti digitali elementari, chiamati porte logiche
- Le porte logiche sono i circuiti minimi per l'elaborazione di segnali binari
- L'elemento funzionale fondamentale per la costruzione di porte logiche è il transistor

Tipi di porte logiche

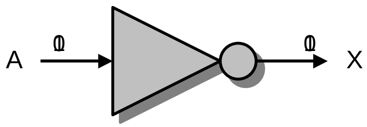
- Classificazione per modo di funzionamento: porta NOT, porta porta AND, porta OR (sono le porte logiche fondamentali)
- Classificazione per numero di ingressi: porte a 1 ingresso, porte a 2 ingressi, porte 3 ingressi, e così via ...

Animazione

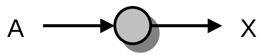
Fine

Porta NOT
(invertitore, negatore)

Simbolo funzionale



(a 1 ingresso)



simbolo semplificato

Tabella delle verità

A	X
0	1
1	0

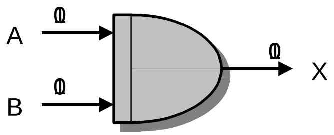
L'uscita vale 1 se e solo se l'ingresso vale 0

Animazione

Fine

Porta AND

Simbolo funzionale



(a 2 ingressi)

L'uscita vale 1 se e solo se entrambi gli ingressi valgono 1

Tabella delle verità

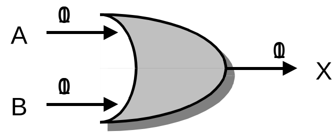
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Animazione

Fine

Porta OR

Simbolo funzionale



(a 2 ingressi)

L'uscita vale 1 se e solo se almeno un ingresso vale 1

Tabella delle verità

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Generalizzazioni

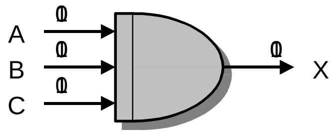
- Alcuni tipi di porte a 2 ingressi si possono generalizzare a 3, 4, ecc ingressi
- Le due porte a più ingressi maggiormente usate sono la porta AND e la porta OR
- Tipicamente si usano AND (o OR) a 2, 4 o 8 ingressi (raramente più di 8)

Animazione

Fine

Porta AND a 3 ingressi

Simbolo funzionale



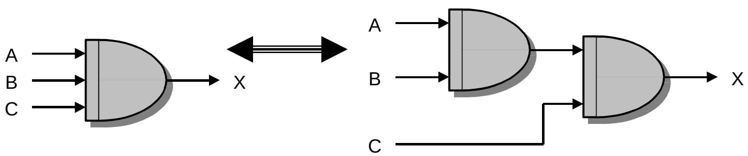
L'uscita vale 1 se e solo se tutti e 3 gli ingressi valgono 1

Tabella delle verità

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

Realizzazione ad albero

- La porta AND a 3 ingressi si realizza spesso come albero di porte AND a 2 ingressi (ma non è l'unico modo)



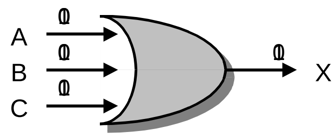
- Nota bene: non tutti i tipi di porte a più di 2 ingressi si possono realizzare come alberi di porte a 2 ingressi (funziona sempre con AND e OR)

Animazione

Fine

Porta OR a 3 ingressi

Simbolo funzionale



L'uscita vale 0 se e solo se tutti e 3 gli ingressi valgono 0

Tabella delle verità

A	B	C	X
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Porte AND e OR a più ingressi

- L'uscita X della porta AND a 3 ingressi vale 1 se e soltanto se tutti e tre gli ingressi A, B e C valgono 1
- L'uscita X della porta OR a 3 ingressi vale 1 se e soltanto se almeno uno tra gli ingressi A, B e C vale 1
- Si generalizza a più ingressi nel modo ovvio ...

Costo di una porta logica

- Il numero di transistor per realizzare una porta dipende dalla tecnologia, dalla funzione e dal numero di ingressi
- Porta NOT: 1 oppure 2 transistor
- Porte AND e OR: 3 oppure 4 transistor
- Altre porte: $\geq$ 4 transistor

Velocità di una porta logica

- La velocità di commutazione di una porta dipende dalla tecnologia, dalla funzione e dal numero di ingressi
- Le porte più veloci (oltre che più piccole) sono tipicamente le porte NAND e NOR a 2 ingressi: possono commutare in meno di 1 nanosecondo (10^{-9} sec, un miliardesimo di sec)

Tecnologie microelettroniche

- Le due tecnologie microelettroniche oggi più usate per la realizzazione di porte logiche sono:
  - a transistor bipolari (o a giunzione), o a transistor BJT (Bipolar Junction Transistor)
  - a transistor a effetto di inversione, o a transistor MOS (Metal Oxide Semiconductor)

Tecnologie bipolari

- Tecnologia TTL (Transistor Transistor Logic): è la tecnologia “storica”, molto usata per anni, ma ora in diminuzione; funziona con alimentazione a 5 Volt
- Tecnologia ECL (Emitter Couple Logic): è una tecnologia costosa, ad alto consumo di potenza, ma estremamente veloce (usata nei supercalcolatori)

Tecnologie complementari

- Tecnologia CMOS (Complementary Metal Oxide Semiconductor): è le tecnologia DI GRAN LUNGA OGGI DOMINANTE per la realizzazione di porte logiche, perché permette
  - dimensioni ridottissime
  - basso consumo
  - basso costo

Algebra di Boole, Funzioni e Reti Combinatorie

Algebra di Boole

- L’algebra di Boole (dal suo inventore G. Boole) serve a descrivere matematicamente i circuiti digitali (o circuiti logici)
- Componenti dell’algebra di Boole:
  - Operatori booleani
  - Regole di trasformazione ed equivalenza tra operatori booleani

Operatori booleani

Nome	Operazione	Porta associata
Inversione	X = /A	Porta NOT
Somma logica	X = A + B	Porta OR
Prodotto logico	X = A B	Porta AND

A, B e X sono variabili booleane
A, B, X $\in \{0, 1\}$
Il prodotto ha precedenza sulla somma

Operatori booleani

- Gli operatori NOT, AND e OR sono quelli fondamentali
- Ogni operatore booleano corrisponde a una porta logica
- Si può rappresentare il funzionamento di un operatore booleano tramite la tabella delle verità della porta associata

Operatori booleani

Somma	Prodotto	Inversione
0 + 0 = 0	0 0 = 0	/0 = 1
0 + 1 = 1	0 1 = 0	/1 = 0
1 + 0 = 1	1 0 = 0	
1 + 1 = 1	1 1 = 1	

- Sono le tabelle delle verità della porta logica OR, AND e NOT, rispettivamente

Proprietà degli op. booleani

Legge	Prodotto logico (AND)	Somma logica (OR)
Identità	1 A = A	0 + A = A
Elemento nullo	0 A = 0	1 + A = 1
Idempotenza	A A = A	A + A = A
Inverso	A /A = 0	A + /A = 1
Commutativa	A B = B A	A + B = B + A
Associativa	(A B) C = A (B C)	(A + B) + C = A + (B + C)
Distributiva	A + B C = (A + B) (A + C)	A (B + C) = A B + A C
Assorbimento	A (A + B) = A	A + A B = A
De Morgan	/(A B) = /A + /B	/(A + B) = /A /B

Proprietà degli op. booleani

- Alcune proprietà degli operatori booleani somigliano a quelle dell'algebra numerica tradizionale
- Altre sono piuttosto diverse (per esempio la proprietà di assorbimento)!
- Si possono usare le proprietà degli operatori booleani per trasformare espressioni booleane

Funzioni combinatorie

- Una funzione combinatoria (o funzione booleana, o funz. logica) corrisponde a un'espressione booleana, contenente una o più variabili booleane e gli operatori booleani AND, OR e NOT
- Dando dei valori alle variabili booleane della funzione combinatoria, si calcola il corrispondente valore della funzione

Esempio

- F(A, B, C) = AB + /C è una funzione combinatoria a 3 variabili A, B e C
- F(0, 0, 0) = 0 0 + /0 = 0 + 1 = 1
- F(0, 0, 1) = 0 0 + /1 = 0 + 0 = 0
- F(0, 1, 0) = 0 1 + /0 = 0 + 1 = 1
- … (e così via)
- F(1, 1, 1) = 1 1 + /1 = 1 + 0 = 1

Tabella delle verità

- La tabella delle verità è un modo per rappresentare il comportamento di una funzione combinatoria
- La tabella delle verità ha due colonne:
  - colonna degli ingressi, le cui righe contengono tutte le combinazioni di valori delle variabili della funzione
  - colonna dell'uscita, che riporta i corrispondenti valori assunti dalla funzione

Esempio

n = 3 ingressi

colonna ingressi

colonna uscita

$2^n = 2^3 = 8$ righe

# riga	A	B	C	A B + /C	F
0	0	0	0	0 0 + /0	1
1	0	0	1	0 0 + /1	0
2	0	1	0	0 1 + / 0	1
3	0	1	1	0 1 + /1	0
4	1	0	0	1 0 + /0	1
5	1	0	1	1 0 + /1	0
6	1	1	0	1 1 + /0	1
7	1	1	1	1 1 + /1	1

(per comodità nella colonna centrale è riportato anche il calcolo)

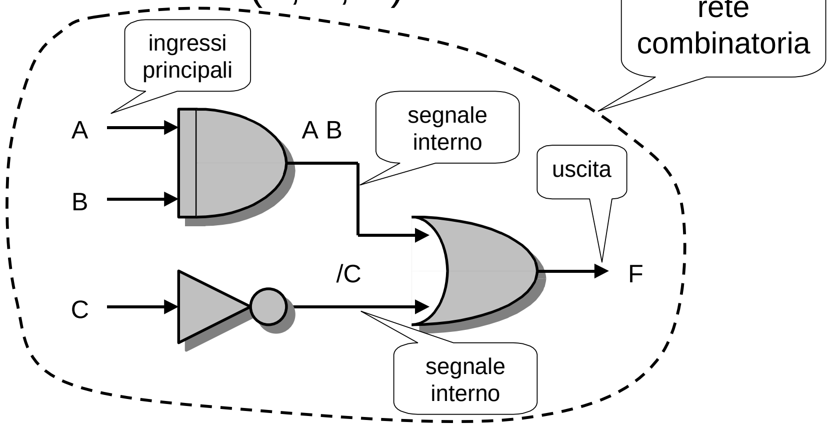
Rete combinatoria

- A ogni funzione combinatoria, data come espressione booleana, si può sempre associare un unico circuito digitale, formato da porte logiche, che viene chiamato rete combinatoria
- Gli ingressi della rete combinatoria sono le variabili della funzione
- L'uscita della rete combinatoria emette il valore assunto dalla funzione

Esempio

funzione combinatoria

F(A, B, C) = A B + /C



Rete combinatoria

- Una rete combinatoria è un circuito digitale:
  - dotato di $n \geq 1$ ingressi principali e di un'uscita
  - formato da porte logiche AND, OR e NOT
  - e privo di retroazioni
- Eventualmente, una rete combinatoria può anche essere formata da porte logiche di altro tipo

Simulazione circuitale

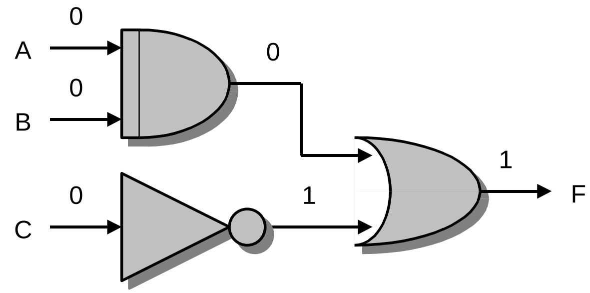
- La tabella delle verità di una rete combinatoria può anche essere ricavata per simulazione del funzionamento circuitale della rete combinatoria stessa
- Per simulare il funzionamento circuitale di una rete combinatoria, si applicano dei valori agli ingressi, e li si propaga lungo la rete fino all'uscita

Animazione

Fine

Simulazione circuitale

(corrisponde alla riga 0 della tabella)



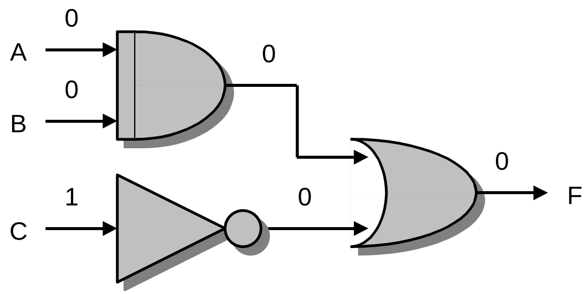
Risultato della simulazione: F(0, 0, 0) = 1

Animazione

Fine

Simulazione circuitale

(corrisponde alla riga 1 della tabella)



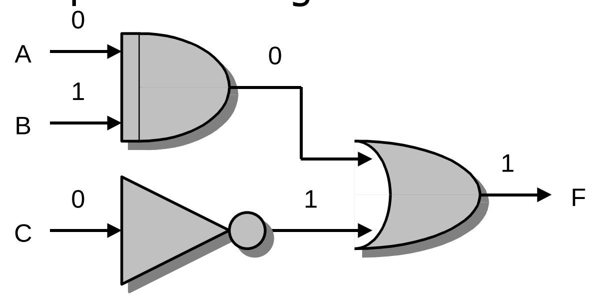
Risultato della simulazione: F(0, 0, 1) = 0

Animazione

Fine

Simulazione circuitale

(corrisponde alla riga 2 della tabella)



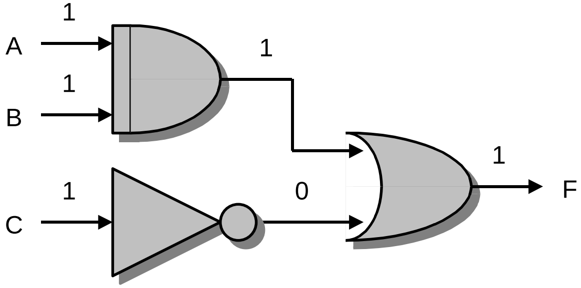
Risultato della simulazione: F(0, 1, 0) = 1

Animazione

Fine

Simulazione circuitale

(corrisponde alla riga 7 della tabella)



Risultato della simulazione: F(1, 1, 1) = 1

Simulazione circuitale

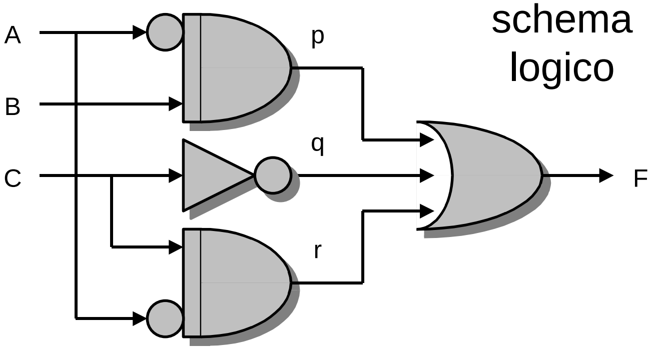
(per comodità è riportato anche il calcolo)

# riga	A	B	C	/A B + /C + /A C	F
0	0	0	0	/0 0 + /0 + /0 0	1
1	0	0	1	/0 0 + /1 + /0 1	1
2	0	1	0	/0 1 + /0 + /0 0	1
3	0	1	1	/0 1 + /1 + /0 1	1
4	1	0	0	/1 0 + /0 + /1 0	1
5	1	0	1	/1 0 + /1 + /1 1	0
6	1	1	0	/1 1 + /0 + /1 0	1
7	1	1	1	/1 1 + /1 + /1 1	0

Analisi di reti combinatorie

Si applicano nomi ai segnali interni: p, q e r

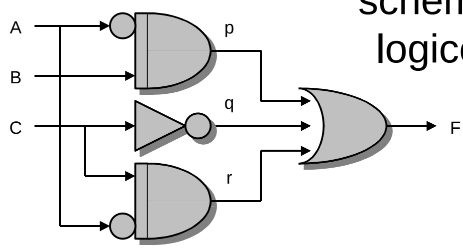
schema logico



Analisi di reti combinatorie

- Si ricavano le espressioni booleane corrispondenti ai segnali interni:
  - p = /A B
  - q = /C
  - r = /A C

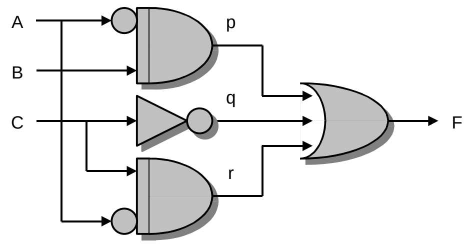
schema logico



Analisi di reti combinatorie

- Si ricava l'uscita come espressione booleana in funzione dei segnali interni:
  - F = p + q + r

schema logico



Analisi di reti combinatorie

- Per sostituzione, si ricava l'uscita come espressione booleana in funzione degli ingressi principali:
  - F = p + q + r
  - F(A, B, C) = /A B + /C + /A C
- L'espressione booleana così trovata ha una struttura conforme allo schema logico di partenza

Sintesi di reti combinatorie

- La sintesi di una rete combinatoria espressa come tabella delle verità, consiste nel ricavare lo schema logico (il circuito digitale) che calcola la funzione combinatoria
- In generale, per una data tabella delle verità possono esistere più reti combinatorie (la soluzione al problema di sintesi non è dunque unica)

Sintesi di reti combinatorie

- Esistono svariate procedure di sintesi di reti combinatorie, che differiscono per:
  - Complessità della procedura di sintesi
  - Ottimalità della rete combinatoria risultante, per dimensioni e velocità
- Una tecnica di sintesi semplice e universale, benché non sempre ottimale, è la sintesi in 1ª forma canonica, o come somma di prodotti

Sintesi in 1ª forma canonica
(o sintesi come somma di prodotti)

- Scrivere la tabella delle verità, a $n \geq 1$ ingressi, della funzione da sintetizzare
- Introdurre n invertitori per generare la negazione di ogni segnale di ingresso principale
- Introdurre una porta AND a n ingressi per ogni 1 presente nella colonna dell'uscita della tabella delle verità

Sintesi in 1ª forma canonica
(o sintesi come somma di prodotti)

- Collegare gli ingressi delle porte AND così introdotte agli ingressi principali, in forma diretta o negata, in modo appropriato
- Inviare l'uscita di tutte le porte AND a un'unica porta OR, dotata di tanti ingressi quante sono le porte AND così introdotte

Esempio: funz. maggioranza

- Si chiede di sintetizzare (in 1ª forma canonica) una funzione combinatoria dotata di 3 ingressi A, B e C, e di un'uscita F, funzionante come segue:
  - Se la maggioranza degli ingressi vale 0, l'uscita vale 0
  - Se la maggioranza degli ingressi vale 1, l'uscita vale 1

Tabella delle verità

- La tabella delle verità della funzione maggioranza è mostrata a lato
- L'uscita vale 1 se e solo se 2 o tutti e 3 gli ingressi valgono 1 (cioè se e solo se il valore 1 è in maggioranza)

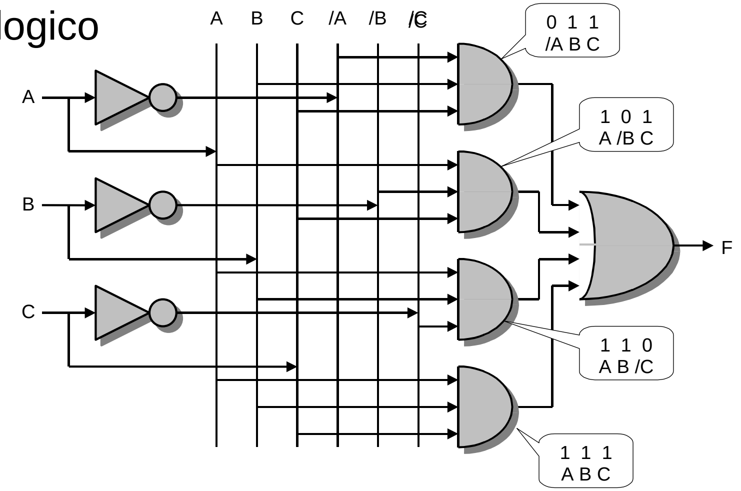
# riga	A	B	C	F
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Animazione

Fine

Rete combinatoria

schema logico



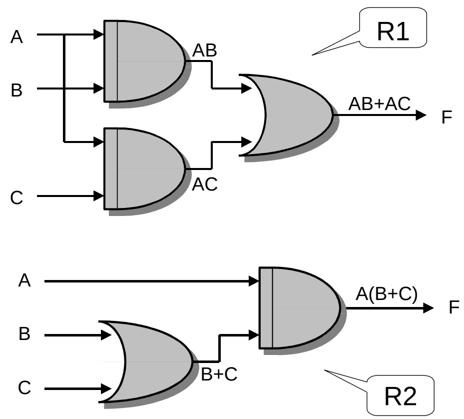
Espressione booleana

- Dallo schema logico della rete combinatoria così sintetizzata, si può ricavare la funzione combinatoria data come espressione booleana
- F(A, B, C) = /A B C + A /B C + A B /C + + A B C
- Nota bene: è una somma di prodotti

Reti combinatorie equivalenti

- Una funzione combinatoria, data come tabella delle verità, può ammettere più reti combinatorie differenti che la sintetizzano
- Reti combinatorie che realizzano la medesima funzione combinatoria si dicono equivalenti
- Esse hanno tutte la stessa funzione, ma struttura (e costo) differente

Due reti equivalenti



F1 = AB + AC

F2 = A(B + C)

Trasformazione:

F1 = AB + AC =

= A(B + C) =

= F2

(prop. distributiva)

Costo di reti combinatorie

- Il costo di una rete combinatoria si valuta in vari modi (criteri di costo):
  - Numero di porte, per tipo di porta e per quantità di ingressi della porta
  - Numero di porte universali (NAND o NOR)
  - Numero di transistor
  - Complessità delle interconnessioni
  - e altri ancora ...

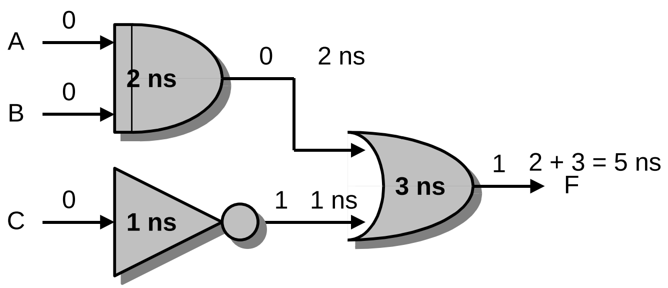
Velocità

- La velocità di una rete combinatoria è misurata dal tempo che una variazione di ingresso impiega per modificare l'uscita della rete (o ritardo di propagazione)
- Per calcolare la velocità di una rete combinatoria, occorre conoscere i ritardi di propagazione delle porte logiche componenti la rete, e poi analizzare i percorsi ingressi-uscita

Animazione

Fine

Velocità



Ritardo totale = 5 ns = 5 10^{-9} sec

Freq. di commutazione = 1 / 5 ns = 200 MHz