



Università degli Studi dell'Insubria
Dipartimento di Scienze Teoriche e Applicate

Architettura degli elaboratori

Il Livello Logico-Digitale

Porte logiche
Algebra di Boole

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

- Il segnale binario è adottato per convenienza tecnica
 - ▶ In linea di principio si potrebbe usare un segnale ternario o a n valori
- Rappresentazione fisica del segnale binario: si usano svariate grandezze fisiche
 - ▶ tensione elettrica (la più usata)
 - ▶ corrente elettrica
 - ▶ luminosità
 - ▶ e altre grandezze fisiche ancora ...

Il segnale binario

- Elaborazione del segnale binario viene fatto da reti logiche
 - ▶ Combinatorie (realizzano funzioni)
 - ▶ Sequenziali (hanno uno stato, o una “memoria”)
- Le reti sono circuiti digitali (o numerici, o logici) composti da porte logiche

- *livello microarchitettura:*
i **circuiti digitali** sono assemblati insieme (e pilotati una *unità di controllo*, che è solo un altro circuito), in una macchina in grado di eseguire *istruzioni macchina* di un dato *instruction set*
- *livello logico:*
Le **porte logiche** vengono assemblate in *circuiti digitali*, che svolgono varie funzioni (di calcolo, di memoria, di controllo...)
- *livello dei dispositivi:*
il **transistor** è l'elemento funzionale fondamentale per la costruzione di *porte logiche*



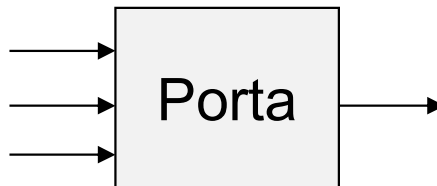
Livello della microarchitettura

Livello logico

Livello dei dispositivi

Porte logiche

- Minuscoli dispositivi dotati di alcuni cavi («wire») di ingresso, e cavo di uscita



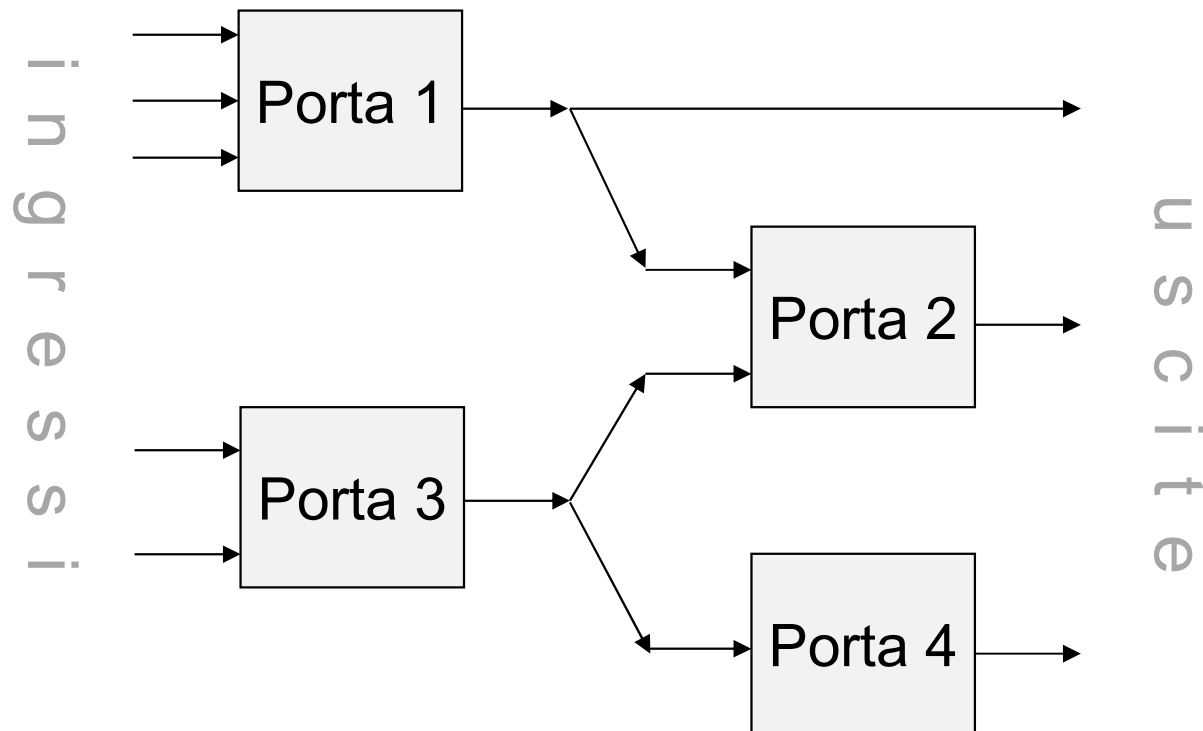
- Funzionamento:
 1. dai cavi di ingresso viene immesso un certo segnale binario (di *input*)...
... e, dopo un certo tempo (brevissimo: frazioni di nanosec) ...
 2. dai cavi di uscita esce un certo altro segnale (elaborato, di *output*)
 - ▶ sia gli input e gli output sono codificati nello stesso modo fisico (esempio con una tensione)
 - ▶ Finché il segnale di input resta invariato, neanche l'output cambia
 - ▶ Quando il segnale di input cambia, dopo un breve tempo il segnale di output cambia (oppure no)
- Esistono molti tipi di porta

Tipi di porte logiche

- Ogni *porta logica* implementa una *funzione logica*
- Classificazione:
per numero di ingressi:
 - ▶ porte a 1 ingresso, (dette anche unarie, o monovariate)
 - ▶ porte a 2 ingressi, (dette anche binarie, o bivariate)
 - ▶ porte a 3 ingressi, (dette anche ternarie, o trivariate)
 - ▶ e così via ...
- Una porta logica a n ingressi implementa una funzione logica a n variabili!
- Classificazione:
per funzione implementata:
porta NOT, porta AND, porta OR, ...

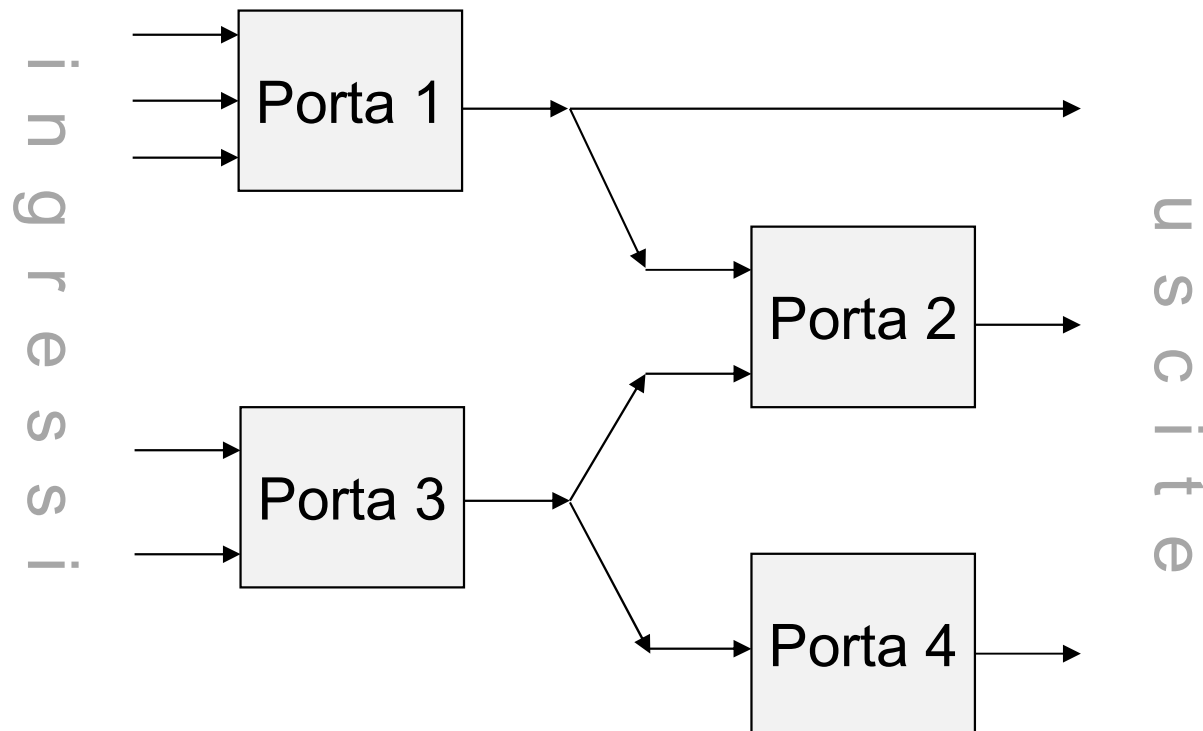
Circuiti digitali (o reti digitali)

- Collegando gli output di una porta logica agli input di un'altra, e così via, costruiremo dei circuiti logici (circuiti digitali, o reti) che *implementano* funzioni a molti input ottenendo così elaborazioni via via più complesse



Circuiti digitali (o reti digitali)

- Un circuito digitale:
 - ▶ è dotato di $n \geq 1$ ingressi e di un'uscita
 - ▶ è formato da porte logiche interconnesse da cavi
 - ▶ l'uscita di una porta è connessa con entrata in una porta o con l'output del circuito



Circuiti digitali: combinatori VS sequenziali

- Due tipi di circuiti:

- ▶ **combinatori**

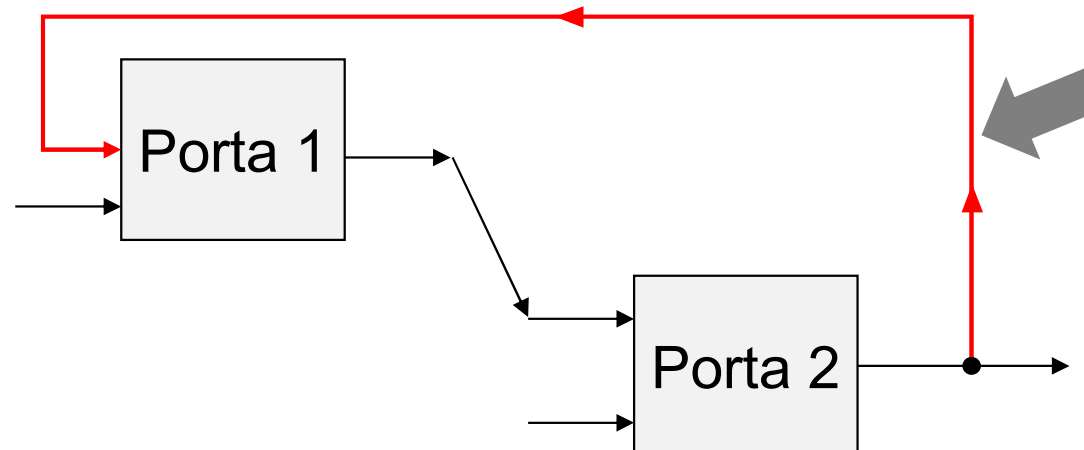
sono privi di retroazioni

- il segnale viaggia dall'input all'output «a senso unico»
 - c'è una gerarchia (un ordinamento parziale) fra le porte
 - niente cicli (aciclico)!

- ▶ **sequenziali**

hanno retroazioni

- *retroazione*: quando il segnale che esce da una porta torna indietro alla stessa porta (anche dopo essere passato da altre porte)
 - esempio:



*per ora,
ci concentriamo solo su questi*

Quali tipi di porte logiche usare?

Porte logiche fondamentali

- Vogliamo usare un insieme di tipi di porte logiche che ci consenta di realizzare qualunque funzione.
 - ▶ Sarebbe anche opportuno che l'insieme fosse piccolo! per ridurre i costi
 - ▶ La teoria ci dice che esistono diversi insiemi siffatti
- Noi useremo (soprattutto) l'insieme: { NOT , AND , OR }
 - ▶ Implementano funzioni logiche molto intuitive
 - ▶ che hanno una lunghissima storia di uso nella logica (da Aristotele in poi!)
 - ▶ L'insieme consente di realizzare qualsiasi funzione
 - ▶ Non è l'insieme più piccolo possibile, ma è comodo da usare
- Pro-memoria: esistono insiemi più piccoli ma ancora sufficienti, come: { NOT, OR } , { NOT, AND } , { NAND } , { NOR }
 - ↑ usatissimo in pratica

Descriviamo alcuni tipi di porte logiche

- Ogni **porta logica** (a n ingressi binari) implementa un **operatore logico** (a n variabili booleane)
- Di ogni porta logica che usiamo, ci interessa:
 - ▶ con quale *simbolo grafico* rappresentarla nei nostri schemi (seguendo delle tradizioni consolidate)
 - ▶ con quale *nome*, e quali *sinonimi*, chiamarla (idem)
 - ▶ quale simbolo usare per l'operatore implementato (idem)
 - ▶ e soprattutto ...
quale operatore logico implementa,
cioè **quale sia la funzione logica corrispondente**

Come descrivo una funzione logica

- *Problema:* come faccio a descrivere una data **funzione logica f** ?

$$y = f(x)$$

o più in generale

$$y = f(x_1, x_2, x_3, \dots)$$

- *Risposta:* posso **tabellarla!**
 - ▶ cioè riportare esaustivamente cosa faccia $f(x_1, x_2, x_3, \dots)$ per ogni combinazione possibile di x_1, x_2, x_3, \dots
 - ▶ nota:
lo posso fare perché esiste solo un numero finito di valori possibili:
 x_1 vale 0 oppure 1
 - ▶ se ho n valori, ho solo 2^n combinazioni da specificare
 - ▶ questa tabella è detta **tabella delle verità**

Porta NOT (invertitore, negatore)

Simbolo funzionale

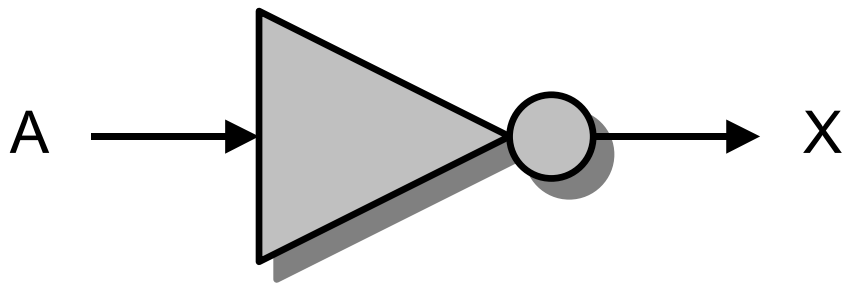
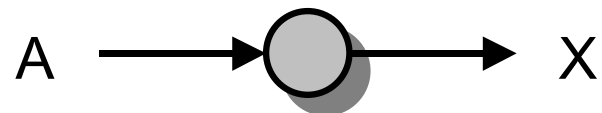


Tabella delle verità

A	X
0	1
1	0

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

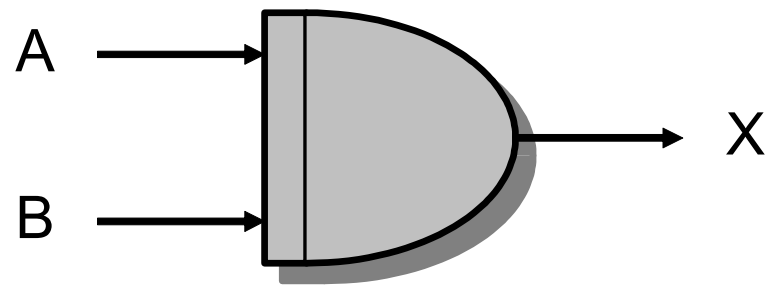
L'uscita vale 0 se
e solo se
l'ingresso vale 1



simbolo semplificato

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

Porta OR

Simbolo funzionale

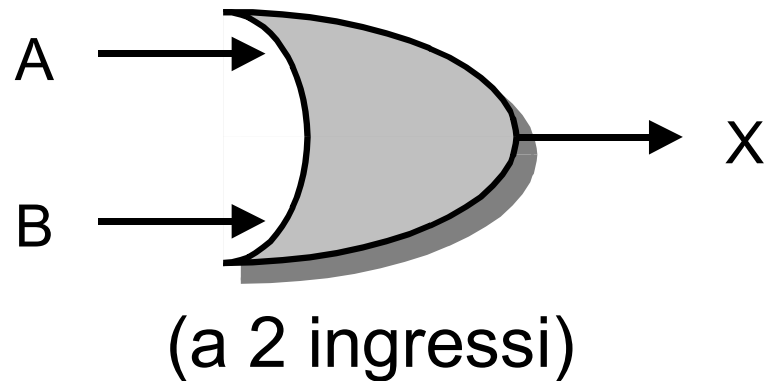


Tabella delle verità

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

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

È un "or" inclusivo