# Introduzione al Progetto Logico di Sistemi Digitali

# Progetto Logico di Sistemi Digitali

- Pervasività hardware digitale
  - Calcolatori
    - ALU, CPU, memorie, I/O, bus...
  - Reti di comunicazione
  - Sistemi dedicati
    - Cellulari, macchine fotografiche, elettrodomestici, autoveicoli, aeromobili...

- Competenze richieste per la progettazione
  - Software e hardware

# Circuito Digitale

- Circuito digitale = Rete Logica

1111010101010011011
1001011100001...

INGRESSI

CIRCUITO DIGITALE
(componenti elettroniche di base interconnesse)

0100101010100010101
010101010010 10...

USCITE

Assumono solo valori <u>discreti</u>
(livelli predefiniti di tensione)

# Livelli di Progetto

1. Livello comportamentale
   - Relazione ingressi/uscite (es: "Uscita = Ingresso + 1")
2. Livello logico
   - Struttura in termini di porte logiche (AND, OR, NOT) e loro interconnessione
3. Livello circuitale
   - Struttura in termini di componenti elettronici (transistor)

- (1) → (2): SINTESI LOGICA, questo corso
- (2) → (3): si usano strumenti automatici

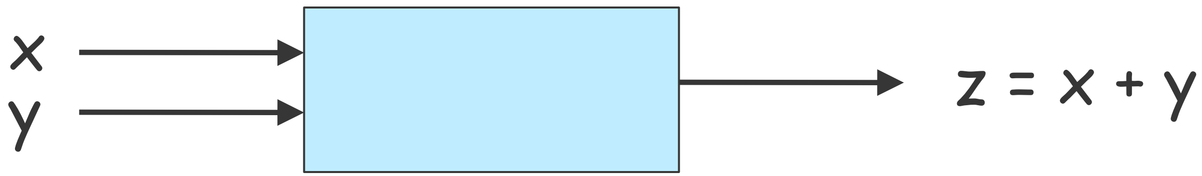
# Progettazione Logica

- Sintesi Logica
  - data una descrizione comportamentale (relazione ingressi/uscite)
  - ottenere una descrizione a livello logico che la realizzi

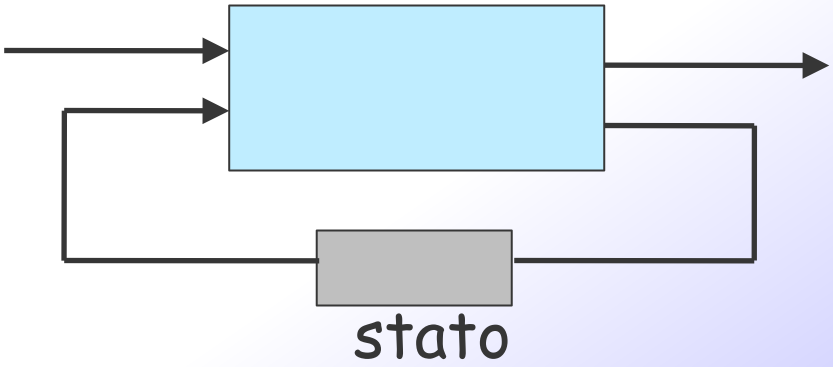
- Problematiche:
  1. Descrizione
     - Come descrivere il sistema a livello comportamentale?
     - Come descrivere il sistema a livello logico?
  2. Ottimizzazione
     - Come scegliere tra più sintesi funzionalmente equivalenti?

# Modelli di reti logiche

- Reti Combinatorie
  - Le uscite dipendono solo dagli ingressi (es: addizionatore)




- Reti Sequenziali
  - Le uscite dipendono sia dagli ingressi che dallo stato interno della rete (es: semaforo: Verde → Giallo → Rosso…)



# Descrizione di Reti Combinatorie

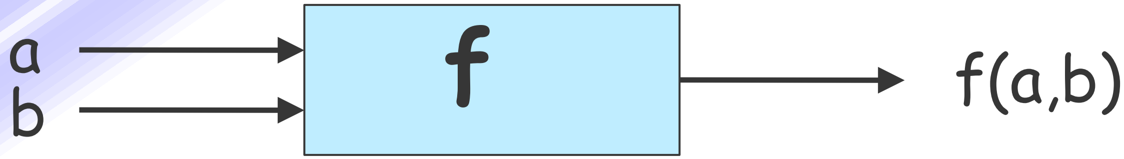
- I segnali di ingresso e di uscita di una rete logica corrispondono a due livelli di tensione distinti
  - Livello “alto”, identificato per comodità con 1
  - Livello “basso”, identificato per comodità con 0

- 0 e 1 sono cifre binarie, ovvero “binary digit” (bit)
- Dunque, una rete combinatoria può essere descritta a livello comportamentale come
  - Una funzione binaria f, che
  - prende in ingresso un insieme di bit
  - restituisce in uscita un insieme di bit

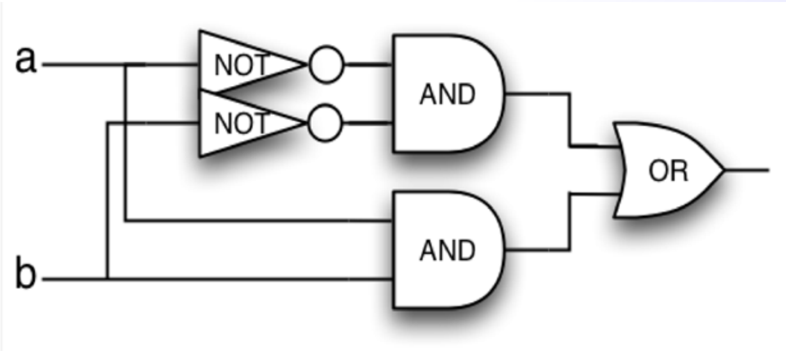
# Descrizione di Reti Combinatorie

- Le funzioni binarie possono essere descritte in modi diversi:
  - Linguaggio naturale
  - Tabelle di verità
  - Espressioni logiche nell'Algebra di Boole
  - Circuiti logici (livello logico)
- Ciascuno di questi è comodo per un particolare scopo
- Sintesi logica:
  - Linguaggio Naturale → Tabelle di Verità → Espressione Booleana

# Esempio




- Linguaggio naturale: "f(a,b) vale 1 solo se gli ingressi sono uguali, vale 0 altrimenti"
- Algebra di Boole: "(a * b) + (a' * b')"
- Tabella di Verità: (vedremo più avanti il significato...)
- Circuito logico:




| a | b | f(a,b) |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

# Descrizione di Reti Sequenziali

- In questo caso non basta specificare la dipendenza delle uscite dagli ingressi

- In aggiunta, è necessario anche specificare
  - La dipendenza delle uscite dallo stato interno
  - Come viene aggiornato lo stato interno

- Descrizioni che verranno utilizzate:
  - Linguaggio naturale
  - Automi di Mealy e Moore
  - Diagramma Stati/Transizioni
  - Tabella degli Stati
  - Circuiti Logici (livello logico)

# Ottimizzazione

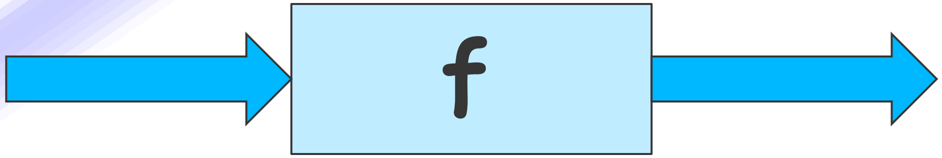
- Ad una descrizione a livello comportamentale C, corrispondono in generale più sintesi logiche possibili S1, S2, S3…
- Come scegliere tra queste?
- Si introducono vincoli non funzionali espressi in termini di cifre di merito, ad esempio:
  - prestazioni (tempo di risposta)
  - dimensioni del circuito
  - costo (es: numero di porte logiche utilizzate)
  - consumo energetico
- Si sceglie la sintesi S che ottimizza la cifra di merito prescelta

# Ottimizzazione (2)

- Reti combinatorie
  - La cifra di merito usata è molto spesso il numero di porte logiche, una volta fissato il numero di livelli di attraversamento del segnale (es: due)
    - Tiene conto sia delle prestazioni (numero di livelli) sia del costo e delle dimensioni (numero di porte)
- Reti sequenziali
  - La cifra di merito è spesso il numero di possibili stati interni
    - Minimizzare il numero di stati porta ad un miglioramento delle reti logiche e ad una riduzione della memoria che viene impiegata nel circuito logico risultante

# Introduzione a Tabelle di Verità, Espressioni Booleane e Porte Logiche

# Tabella di Verità




- Sia f: $\{0,1\}^n \rightarrow \{0,1\}$
- La funzione booleana corrispondente ad una rete combinatoria con n ingressi ed 1 uscita

- Una tabella di verità (TV) è una descrizione di f che specifica, per ogni possibile combinazione dei $2^n$ ingressi, l'uscita corrispondente di f

# Esempio 1

- Linguaggio naturale
  - "f(a,b) vale 1 se e solo se a e b sono uguali"
- Tabella di Verità per f

| Ingresso "a" | Ingresso "b" | Uscita: f(a,b) |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

# Funzioni a più uscite

- $f: \{0,1\}^2 \rightarrow \{0,1\}^2$ funzione a due ingressi e due uscite

- $f(a,b) = (c,d)$, dove
  - 'c' vale 1 se e solo se 'a' e 'b' sono uguali
  - 'd' vale 1 se e solo se 'a' è diverso da 'b' "

| a | b | c | d |
|---|---|---|---|
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 |

# Discussione

- Qualisiasi funzione booleana (rete combinatoria) può essere descritta tramite una Tabella di Verità
- Problema:
  - Al crescere del numero di ingressi, la dimensione della TV cresce molto rapidamente
- Esempio:
  - n=4 ingressi (a, b, c, d) definita come:
  - "f(a,b,c,d)=1 se e solo se esattamente due dei quattro ingressi valgono 1"

# Esempio 2...

| a | b | c | d | f(a,b,c,d) |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 0 |

# ...Esempio 2 (continua)

| a | b | c | d | f(a,b,c,d) |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

# Osservazioni

- Nella TV, in ciascuna riga, gli ingressi codificano un numero in forma binaria crescente. Nell'esempio: 0, 1, 2, ..., 15.
  - Per n=2 ingressi ci sono 4 righe, per n=3 ingressi ci sono 8 righe, per n=4 ci sono 16 righe...
- In generale, per n ingressi ci sono $2^n$ righe
  - La dimensione della TV cresce esponenzialmente con n
  - Quando n è grande, la TV diventa ingestibile
    - Es: con n=20, la TV ha $2^{20}$ > 1 milione di righe
- Si deve ricorrere a dei metodi di descrizione alternativi
  - Espressioni booleane e porte logiche

# Espressioni booleane

- Espressioni costruite a partire da tre operatori logici fondamentali: AND, OR, e NOT

- Si dicono logici perché interpretano valori binari 0 e 1 come valori di verità logica
  - 0 corrisponde a "falso"
  - 1 corrisponde a "vero"

- AND e OR hanno due ingressi e un'uscita (sono operatori binari)
- NOT ha un solo ingresso e un'uscita (è un operatore unario)

- Questi tre operatori corrispondono a tre funzioni booleane molto semplici, descritte in seguito

# AND (a*b)

- a*b=1 se e solo se a=b=1

- Signifgicato logico:
  - '(a AND b)' è vero <u>se e solo se</u> sia 'a' che 'b' sono veri

- TV:

| a | b | a*b |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 1 | 1 |
| 1 | 0 | 0 |

# OR (a+b)

- a+b=1 se e solo se a=1 oppure b=1

- Signifgicato logico:
  - '(a OR b)' è vero <u>se e solo se</u> 'a' è vero oppure 'b' è vero
- TV:

| a | b | a+b |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 1 | 1 |
| 1 | 0 | 1 |

- NB: l'OR non è esclusivo! Se a=b=1, allora a+b=1

# NOT (a')

➢ a'=1 se e solo se a=0

- Signifgicato logico:
  - '(NOT a)' è vero sse a è falso (negazione)
- TV:

| a | a' |
|---|---|
| 0 | 1 |
| 1 | 0 |

# Altre porte logiche

- Oltre a AND, OR, e NOT sono usati altri operatori logici fra cui: XOR, NAND NOR
- Come AND e OR XOR, NAND e NOR hanno due ingressi e un'uscita (sono operatori binari)

| a | b | XOR(a,b) | NAND(a,b) | NOR(a,b) |
|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |

# Esempio

➢ f(a,b)=1 se e solo se a e b sono uguali

- Espressione booleana $f(a,b) = (a * b) + (a' * b')$
- Interpretazione: f vale 1 se...
  - sia a che b sono entrambi uguali a 1 (quindi a*b vale 1)
  - oppure (+)
  - sia a che b sono entrambi uguali a 0 (quindi a'*b' vale 1)

# Notazione

- Per semplificare la notazione, stabiliamo alcune regole

- NOT lega più di AND e OR, mentre AND lega più di OR
  - a*b + a'*b' = (a*b) + ( (a') * (b') )
- L'AND (*) si può omettere
  - ab+c = (a*b)+c
- Il NOT (') si scrive anche "!"
  - !(a+b) = (a+b)'
- AND e OR godono della proprietà associativa
  - a*b*c = a*(b*c) = (a*b)*c

- NB: Naturalmente, queste operazioni + e * non hanno nulla a che vedere con le operazioni aritmetiche somma e moltiplicazione, che vengono indicate con gli stessi simboli!

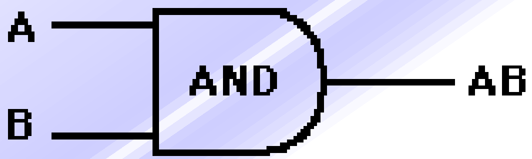
# Esempio 3

- f(a,b,c,d)=1 se e solo se esattamente due tra a, b, c e d sono uguali a 1
- Espressione booleana
  - f(a,b,c,d) = (a*b+ a*c + a*d + b*c + b*d + c*d) *
  - (a*b*c + a*b*d + a*c*d + b*c*d + a*b*c*d)'
- Interpretazione
  - almeno due degli ingressi sono veri (a*b + a*c + a*d + b*c + b*d + c*d)
  - e (*)
  - non ci sono tre o quattro ingressi veri (a*b*c + a*b*d + a*c*d + b*c*d+ a*b*c*d)'

# Porte Logiche

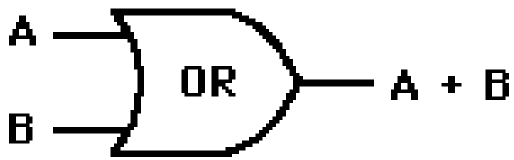
- Sono blocchi di costruzione fondamentali di ogni circuito logico
- Una volta connesse tra loro, varie porte logiche formano un circuito logico
- Le tre porte logiche più comuni realizzano le funzionalità dei tre operatori logici fondamentali:
  - AND
  - OR
  - NOT

# Porte Logiche (2)



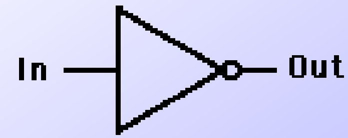

| A | B | Out |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

The AND operation will be signified by AB or $A \cdot B$. Other common mathematical notations for it are $A \wedge B$ and $A \cap B$, called the intersection of A and B.



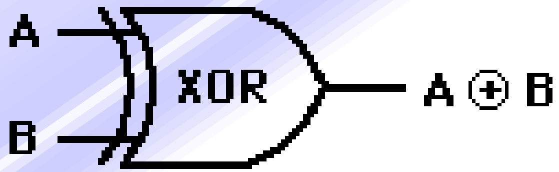

| A | B | Out |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

The OR operation will be signified by $A + B$. Other common mathematical notations for it are $A \vee B$ and $A \cup B$, called the union of A and B.

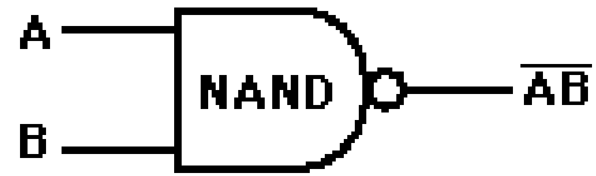



| In | Out |
|---|---|
| 0 | 1 |
| 1 | 0 |

# Porte Logiche XOR , NAND, NOR




| A | B | Out |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

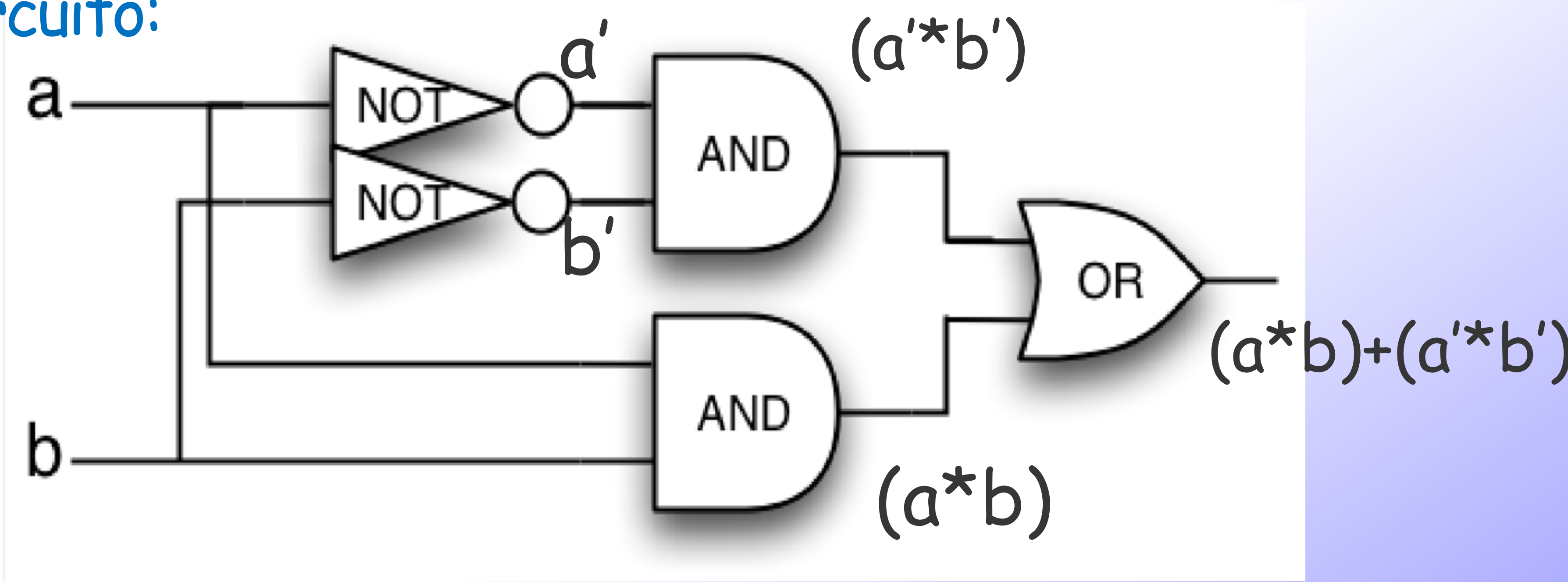



| A | B | Out |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |


| A | B | Out |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

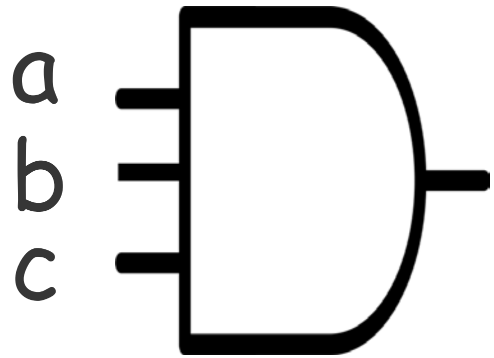
# Circuiti Logici

- Data una espressione booleana, la sua corrispondenza ad un circuito logico è immediata.
- Esempio
  - Espressione: $f(a,b) = a*b + a' * b'$
  - Circuito:



# Porte AND e OR a più ingressi

- Ci sono varianti a n ingressi delle porte AND e OR
- Esempio (n=3)




$d = a * b * c$

$$d = (a*b)*c = a*(b*c)$$

(il modo in cui gli ingressi sono raggruppati non è rilevante! )

# Problemi

- Torniamo al problema della sintesi logica.

1. I tre operatori logici sono sufficienti a descrivere ogni funzione booleana (rete combinatoria)?
2. Se sì, data una TV (o una rappresentazione equivalente), come ricavo automaticamente una corrispondente espressione booleana?
3. Data una espressione booleana, come ricavo una equivalente espressione ottimale, secondo la prescelta cifra di merito?
4. È possibile fare i passi 2 e 3 simultaneamente?

- Per rispondere a queste domande, dobbiamo prima di tutto approfondire lo studio dell'Algebra di Boole

# Esercizio

- Disegnare il circuito logico che corrisponde all'espressione booleana
- $f(a,b,c,d) = (a*b + a*c + a*d + b*c + b*d + c*d) *$
- $* ( a*b*c + a*b*d + a*c*d + b*c*d)'$
- Usare porte AND e OR a due o più ingressi, secondo la necessità

# Soluzione