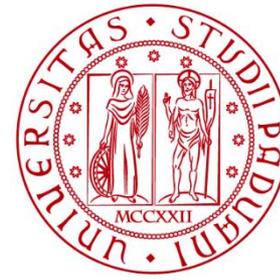




DEI
DIPARTIMENTO DI
INGEGNERIA DELL'INFORMAZIONE



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Sistemi Digitali

Introduzione ai sistemi digitali

Marta Bagatin, marta.bagatin@unipd.it

Corso di Laurea in Ingegneria dell'Informazione
Anno accademico 2022-2023

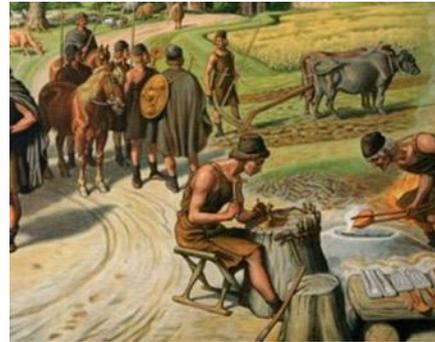
Introduzione ai sistemi digitali

L'era dell'Informazione

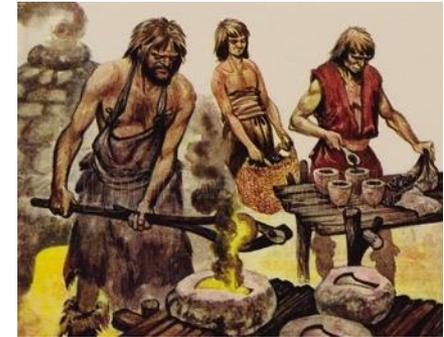
Età della pietra



Età del bronzo



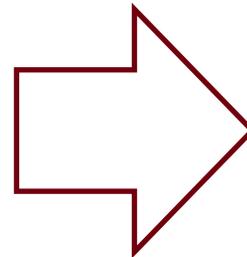
Età del ferro



Machine age (1880 -19xx)

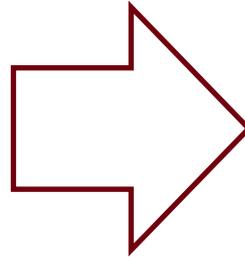


Information age (20xx)



La rivoluzione digitale

Machine age (1880 - 19xx)



Information age (20xx)



- Alla fine degli anni '80, meno dell'1% dell'informazione mondiale era in formato digitale, mentre nel 2007 lo era in una percentuale pari al 94%

Cos'è un sistema digitale?

- Digital: «made with digits» (= **cifra**). «Digit» deriva dal latino «digitus» (= dita), il metodo più antico di contare i numeri è usare le dita



- Un sistema digitale **immagazzina, sposta e processa informazioni digitali**, cioè **opera su elementi discreti**
 - Ogni **insieme con un numero finito di elementi** contiene elementi discreti (esiste un intorno di ogni punto dell'insieme in cui non cadono altri elementi dell'insieme)
 - Es.: le 10 dita delle mani, le 26 lettere dell'alfabeto, le 108 carte di un mazzo da gioco, il numero di persone in una stanza...

Segnali digitali vs. segnali analogici

- Un **sistema digitale opera su segnali digitali**
- Le **grandezze del mondo reale sono continue** (la velocità di un corpo, l'intensità del suono, la pressione atmosferica, ...)
- Le grandezze continue sono descritte da **segnali analogici**, cioè possono assumere **tutti i valori compresi in un intervallo finito**
- Per rappresentare le **grandezze in modo digitale** abbiamo bisogno di digitalizzarle, cioè **renderle discrete**
 - Es: In un termostato digitale che controlla un sistema di riscaldamento, la temperatura di una stanza (segnale continuo) è rappresentata da valori discreti, per es. i numeri interi compresi tra 0 e +40

Perché studiare i sistemi digitali?

- Negli anni '60, la maggior parte dei sistemi elettronici erano **analogici**
- Oggi, praticamente tutti i sistemi sono **digitali**, eccetto quelli “at the edge” (al confine), che fanno da interfaccia con le grandezze del mondo reale
 - Vantaggi dei sistemi digitali
 - Non c'è accumulo di rumore
 - Precisione virtualmente arbitraria
 - Lo scaling dei circuiti integrati (legge di Moore) ha permesso un'evoluzione continua e una crescente miniaturizzazione dei sistemi digitali

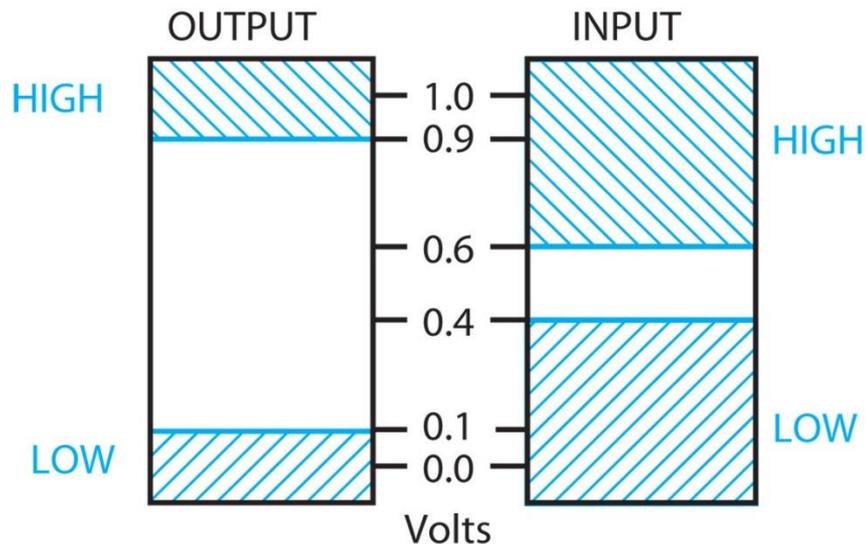
Sistemi digitali e segnali elettrici

- I sistemi digitali vengono realizzati da sistemi elettronici che operano su **tensioni e correnti elettriche**, mediante dispositivi elettronici chiamati **transistor**
 - Intuitivamente: i transistor sono interruttori che lasciano o meno passare un segnale, a seconda della loro configurazione (aperto/chiuso)

Sistemi digitali binari

- I sistemi digitali binari fanno uso di segnali che possono assumere solo **due valori discreti**
 - Tali valori vengono rappresentati con **'1'** e **'0'** (cifre del sistema di numerazione binario), **ON** e **OFF**, oppure **VERO** e **FALSO**. Una **cifra binaria** è detta **bit** (binary digit) di informazione
- Questi due valori corrispondono a grandezze fisiche, cioè a due intervalli di tensione elettrica, detti **valore logico ALTO** e **valore logico BASSO**, che possono essere assunti da ingressi e uscite del sistema digitale

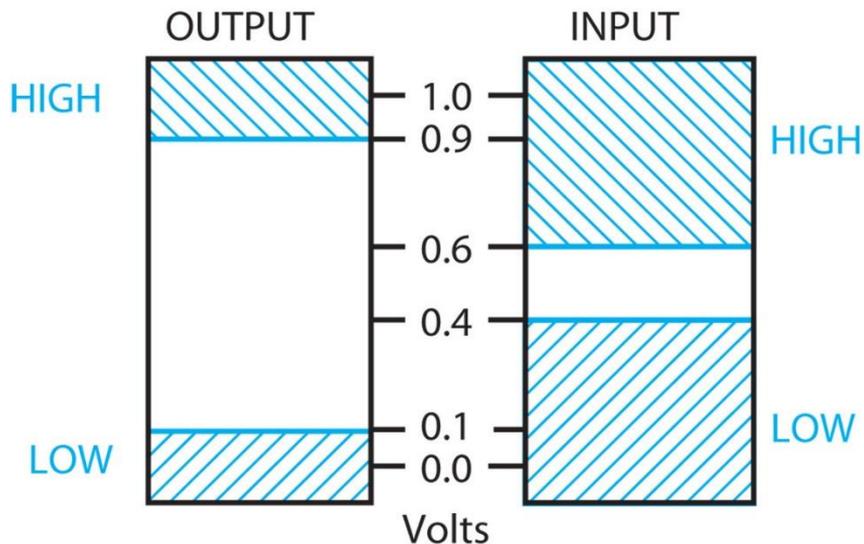
Ingressi e uscite



(a) Example voltage ranges

- Intervalli di tensione che possono assumere gli ingressi e le uscite del sistema
 - **Ingresso** è riconosciuto
 - **alto** se sta tra 0.6 V e 1 V,
 - **basso** se sta tra -0.1 V e 0.4 V
 - **Uscita** è riconosciuta
 - **alta** se sta tra 0.9 V e 1.1 V
 - **bassa** se sta tra -0.1 V e 0.1 V
- L'intervallo dei **valori permessi all'ingresso** è **più ampio** dell'intervallo di **valori permessi in uscita**

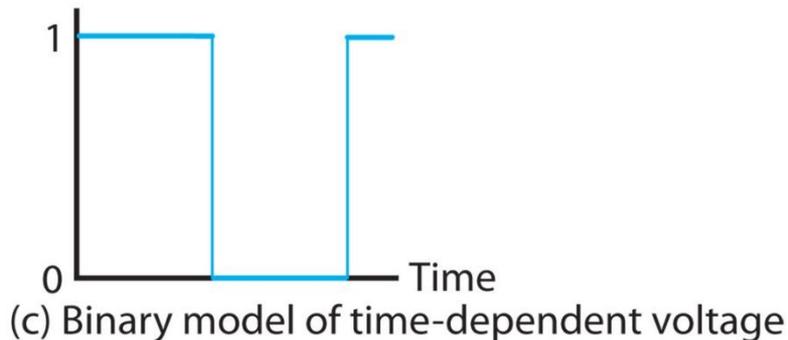
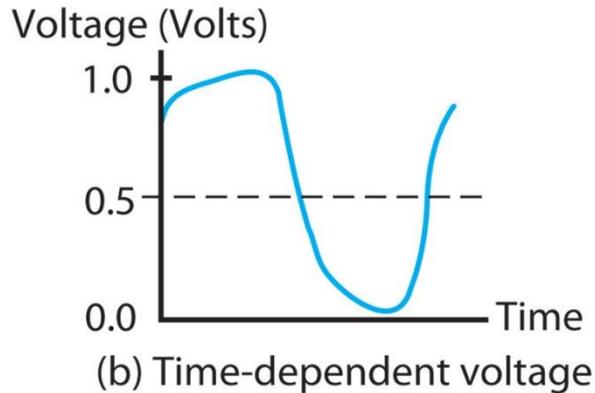
Ingressi e uscite



(a) Example voltage ranges

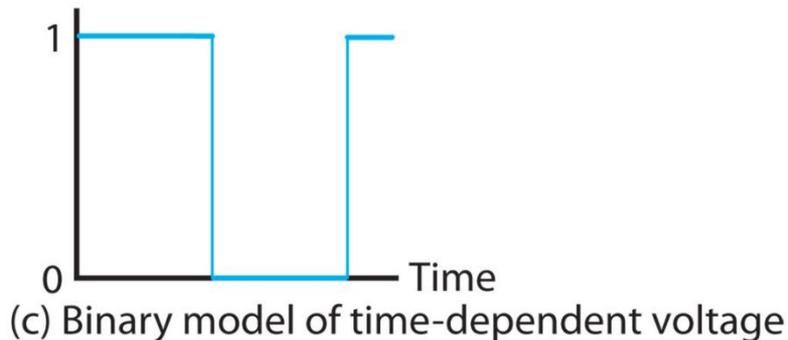
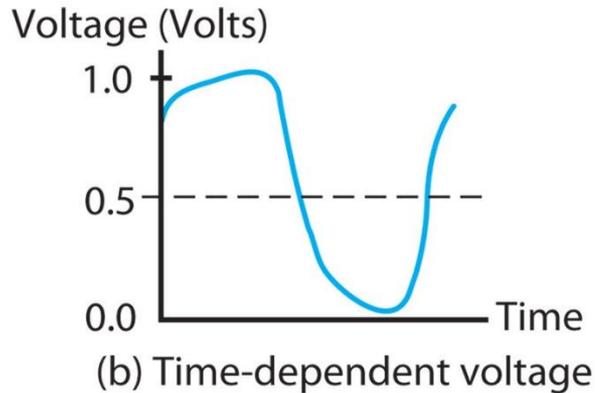
- Il sistema riesce a funzionare correttamente anche in presenza di rumore sovrapposto al segnale di ingresso: **proprietà rigenerativa**
- I circuiti digitali sono più **tolleranti al rumore** rispetto ai circuiti analogici. In presenza di variazioni indesiderate agli ingressi, trasmettono **uscite più 'pulite' e non distorte dal rumore**

Rappresentazione binaria di segnali



- Un segnale continuo come la tensione elettrica può essere **rappresentato con una forma d'onda binaria**: tutti i **valori sopra 0.5 V sono letti come '1' (valore logico alto)**, quelli **sotto 0.5 V sono letti come '0' (valore logico basso)**
- A livello logico, idealizziamo il comportamento di ingressi e uscite di un sistema digitale con quello di cifre binarie che assumono i valori '0' o '1'

Rappresentazione binaria di segnali

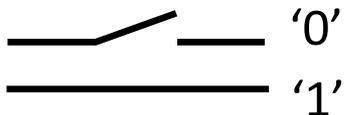


- Questa idealizzazione, in cui si tralasciano gli aspetti elettrici e si evidenziano invece quelli numerico-logici, è un esempio di astrazione
- Si dice che la **descrizione logica avviene ad un livello di astrazione più alto** (=meno dettagliato) **della descrizione elettrica**

Sistemi binari

- Nei sistemi digitali l'informazione è rappresentata sotto forma di gruppi di bit
- Si possono usare **diverse tecniche di codifica** e fare in modo che gruppi di bit rappresentino **non solo numeri, ma anche altri gruppi di simboli discreti**
 - es. codice ASCII: codici di 7 bit usati per codificare i caratteri di una tastiera

Esempi di rappresentazione di grandezze con segnali digitali

- (a) **Informazioni binarie**: segnali binari a singolo bit  '0'
'1'
- Luce accesa o spenta, porta aperta o chiusa

- (b) **Elementi discreti di un insieme**: segnali binari multi-bit, cioè gruppi di segnali binari

- 8 colori possono essere rappresentati con 3 bit

000	white	011	purple
001	red	101	orange
010	blue	110	green
100	yellow	111	black

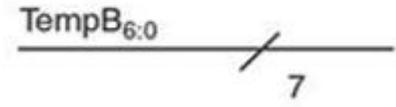
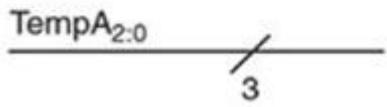
- (c) **Grandezze continue**: possono essere quantizzate e rappresentate con segnali binari multi-bit

- 8 temperature rappresentate con diverse codifiche (rappresentazione scelta dipende dal tipo di applicazione)

000	68	100	76
001	70	101	78
010	72	110	80
011	74	111	82

0000000	68
0000001	70
0000011	72
0000111	74

0001111	76
0011111	78
0111111	80
1111111	82

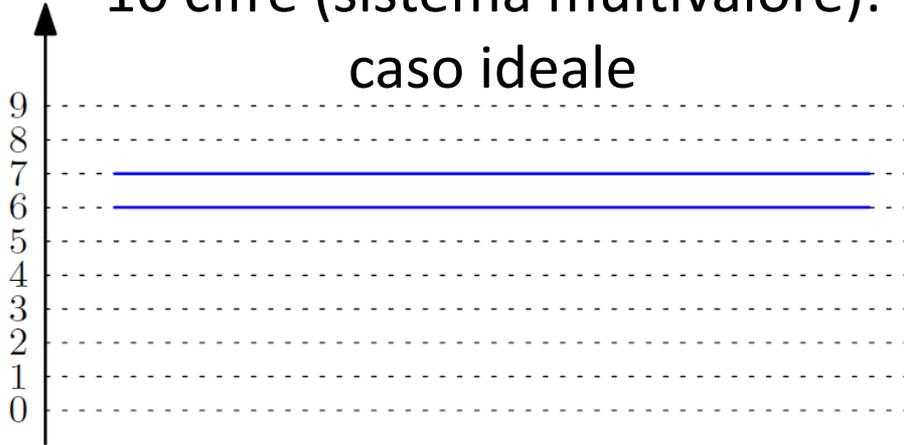


Perché usare sistemi binari?

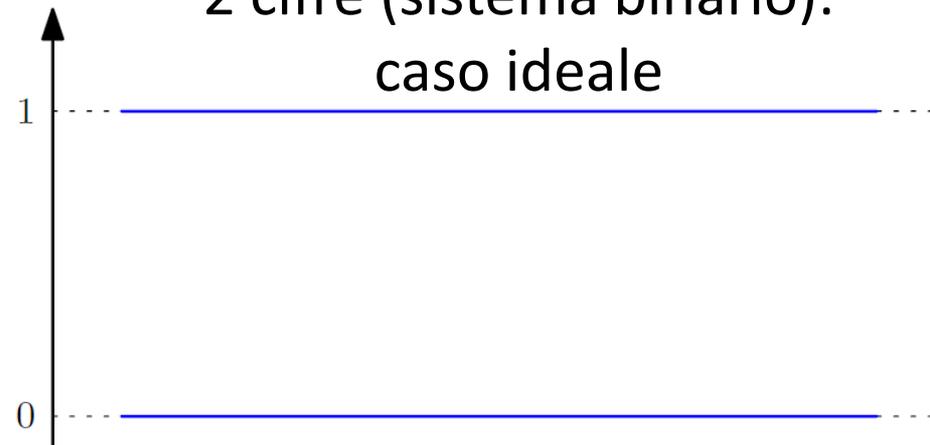
- Un sistema binario ha **una tolleranza al rumore più grande rispetto ad un sistema digitale multivalore**
 - Se dividiamo l'intervallo di tensioni in 10 intervalli anziché in 2 (avremo 10 possibili valori in ingresso e 10 in uscita): a parità di rumore in ingresso, sarà più facile che, a causa del rumore, il valore venga interpretato in modo scorretto, cioè letto in uno slot adiacente
- Quindi i **sistemi digitali binari** sono **più affidabili** rispetto ai sistemi digitali multivalori, garantendo un miglior funzionamento in presenza di rumore all'ingresso o di variazioni non volute

Perché usare sistemi binari?

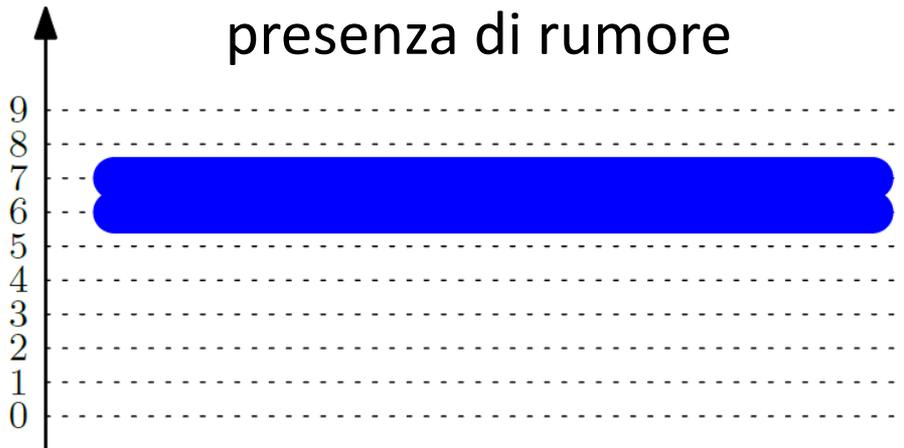
10 cifre (sistema multivalore):
caso ideale



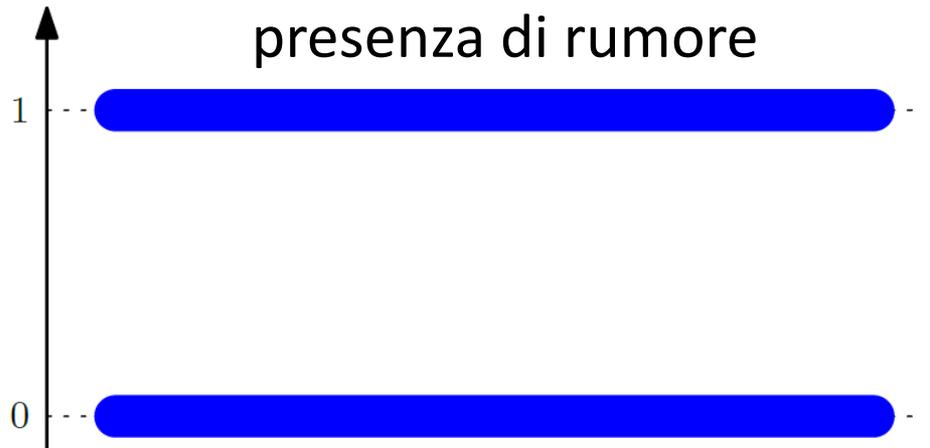
2 cifre (sistema binario):
caso ideale



10 cifre (sistema multivalore):
presenza di rumore



2 cifre (sistema binario):
presenza di rumore

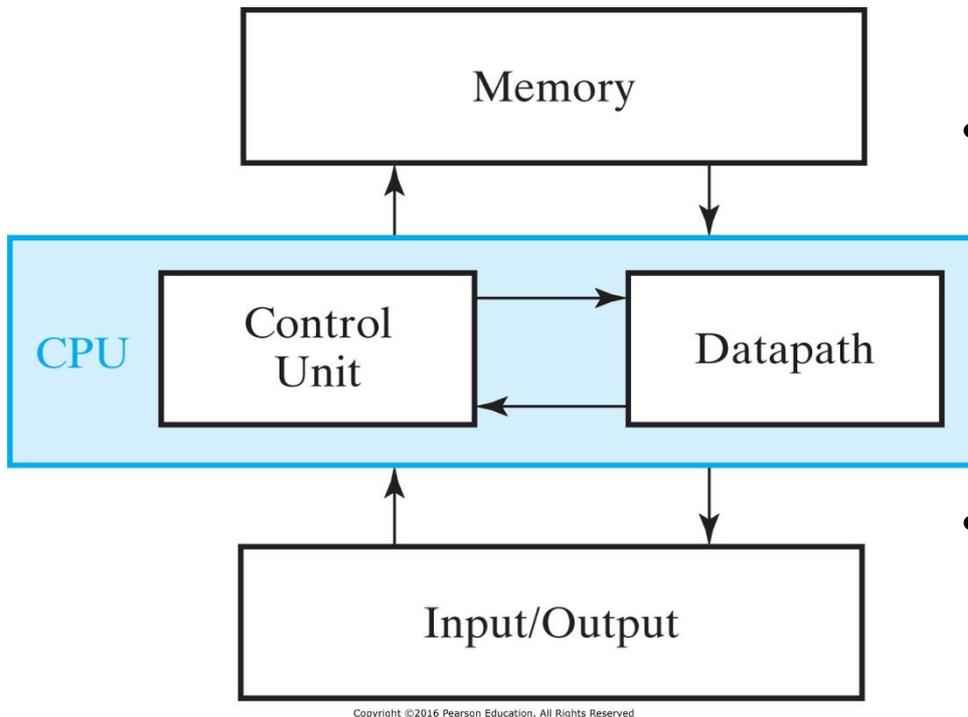


Digital computer e oltre

Uno sguardo all'elaboratore

- Il computer è il **sistema digitale** per eccellenza
- Il termine “**digital computer**” deriva dal fatto che i primi calcolatori erano usati prevalentemente per computazioni numeriche, per cui gli elementi discreti su operavano erano le cifre
- Oggi i computer sono impiegati per l'elaborazione di dati virtualmente in **tutti i campi**
 - Es: applicazioni industriali, bancarie e assicurative, mediche, trasporto, meteorologia, comunicazione, intrattenimento, e progettazione ingegneristica (strumenti CAD – Computer Aided Design, usati in edilizia, meccanica, elettronica, ...)

Schema a blocchi di un elaboratore

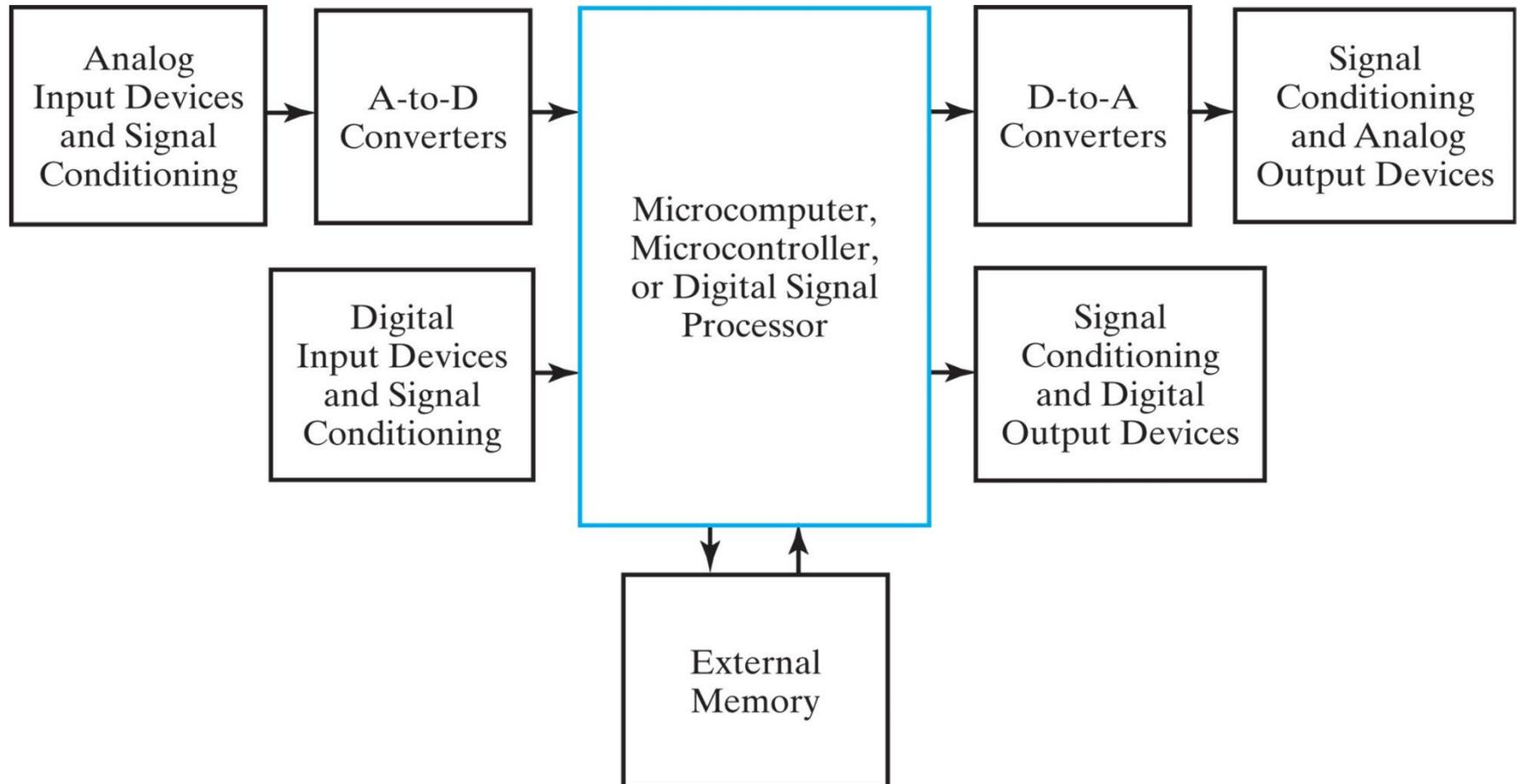


- **Memoria**: immagazzina input, output, dati intermedi, istruzioni
- **CPU: Central Processing Unit**
 - **Control Unit**: supervisiona i flussi di informazioni tra i vari blocchi
 - **Datapath**: esegue operazioni sui dati, secondo quanto specificato dal programma
- **Dispositivi di ingresso** (es: tastiera, scanner, microfono) e **uscita** (es: schermo, stampante, casse) o ingresso/uscita (es: chiavetta USB, DVD)

Oltre il computer

- La definizione di computer può essere estesa ad altri componenti che funzionano in maniera simile
- **Microcomputer o microcontrollori o DSP (Digital Signal Processor)** sono computer meno potenti e più compatti, spesso integrati in un singolo chip di silicio, tipicamente incorporati in sistemi complessi (automobile, telefonino, lavatrice, condizionatore)
- Per questo sono detti microcomputer **embedded** (il contrario di embedded è stand-alone)

Schema a blocchi di un sistema embedded



Dispositivi I/O digitali e analogici: esempi

- **Dispositivi a ingresso digitale**

- Interruttore finecorsa: dispositivo con input digitale binario, può avere o meno una forza applicata in ingresso
- Telecomando: dispositivo con input digitale multivalore, pressione su uno dei suoi 10 tasti

- **Dispositivi a ingresso analogico**

- Termocoppia: sensore che riceve in ingresso una temperatura e produce in uscita una tensione proporzionale alla temperatura
- Quarzo: è composto di cristalli che, in risposta ad una pressione applicata, si deformano e si polarizzano, producendo in uscita una tensione elettrica

- **Dispositivi a uscita analogica**

- Altoparlante: produce onde sonore (continue), convertendo un segnale elettrico

- **Dispositivi a uscita digitale**

- Display a LED: può essere acceso o spento a seconda della tensione applicata
- Relè elettromeccanico: può essere aperto o chiuso, a seconda della tensione applicata

Esempi di sistemi embedded

Application Area	Product
Banking, commerce and manufacturing	Copiers, FAX machines, UPC scanners, vending machines, automatic teller machines, automated warehouses, industrial robots, 3D printers
Communication	Wireless access points, network routers, satellites
Games and toys	Video games, handheld games, talking stuffed toys
Home appliances	Digital alarm clocks, conventional and microwave ovens, dishwashers
Media	CD players, DVD players, flat panel TVs, digital cameras, digital video cameras
Medical equipment	Pacemakers, incubators, magnetic resonance imaging
Personal	Digital watches, MP3 players, smart phones, wearable fitness trackers
Transportation and navigation	Electronic engine controls, traffic light controllers, aircraft flight controls, global positioning systems

Esempi di sistemi embedded

- Un'automobile moderna può contenere svariate **decine di microcontrollori/microcomputer**
- Ognuno di questi si occupa di un blocco di funzionamento dell'auto
 - controllo dell'ABS
 - controllo di stabilità
 - controllo del motore
 - ...

Es: l'automobile moderna



ADAS Driving Growth
 Shift to smaller cars
 Multiple systems x vehicle
 Legislation(NHTSA/EUNCAP)

Courtesy of M. Duncan,
 STMicroelectronics

Source: Strategy Analytics

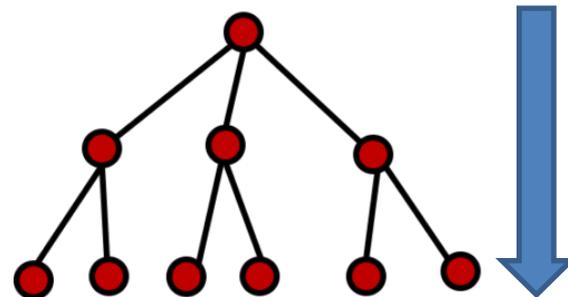


* Percentuale del costo dell'elettronica sul costo totale di un'auto

Livelli di astrazione nella progettazione di sistemi digitali

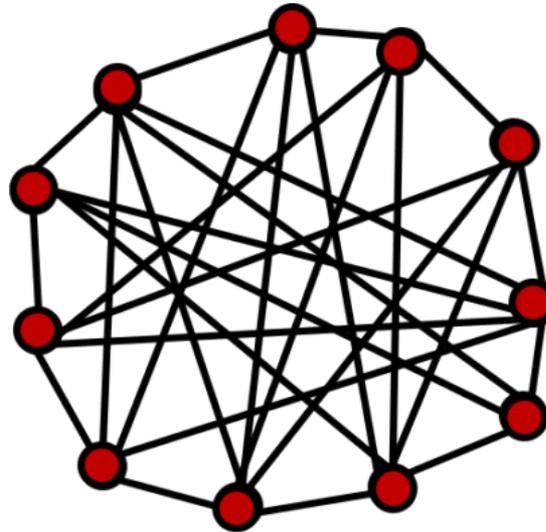
Progettazione di sistemi digitali e livelli di astrazione

- La **progettazione dei sistemi digitali** avviene tipicamente con un **approccio top down**
 - Si inizia dalla **specifica** del sistema ad un livello alto di astrazione (senza occuparsi dei dettagli dei singoli sottoblocchi)
 - Via via si scende, scomponendo successivamente il sistema in **blocchi sempre più piccoli**, fino a raggiungere blocchi abbastanza semplici da poter essere implementati
 - Infine, si costruisce il sistema dall'**unione di questi blocchi**, opportunamente connessi tra loro

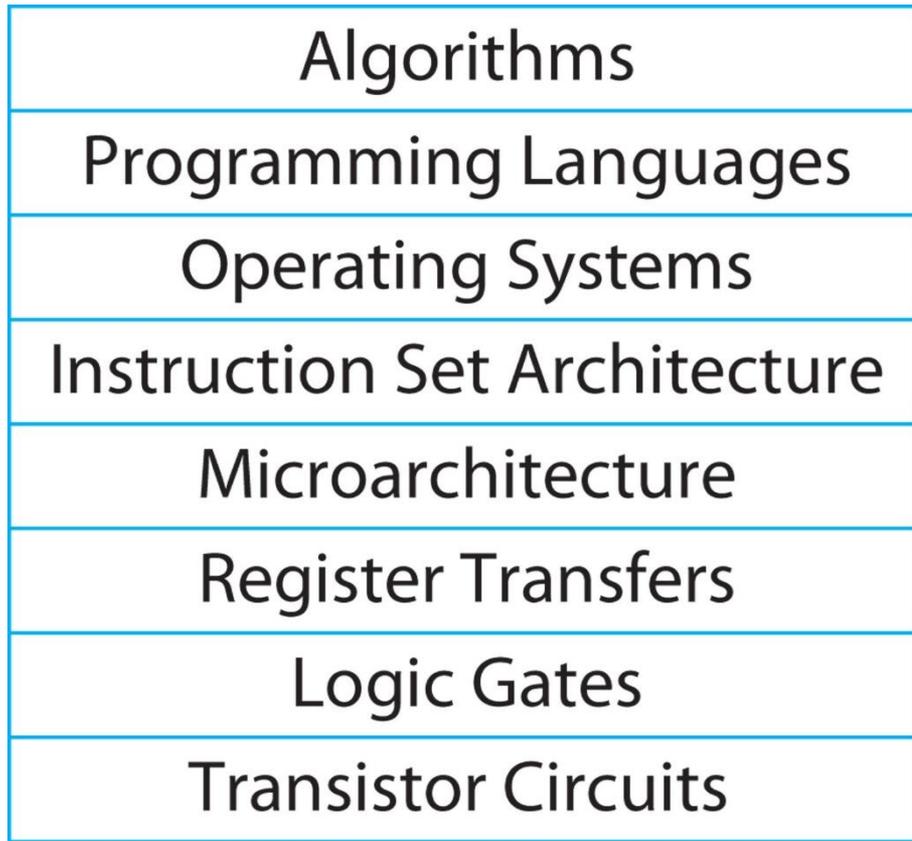


Progettazione di sistemi digitali e livelli di astrazione

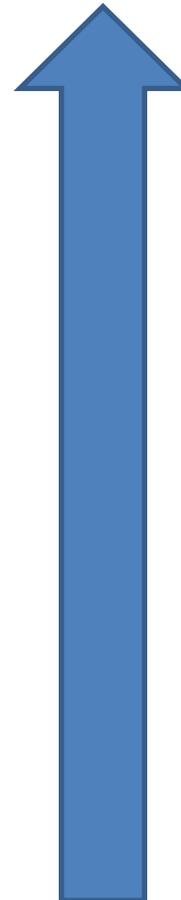
- In questo corso useremo un **approccio bottom up**
- **Partiremo dallo studio e progettazione dei blocchi più semplici, per poi connetterli fino a formare un sistema complesso**



Livelli di astrazione



Copyright ©2016 Pearson Education, All Rights Reserved



Il livello di astrazione aumenta andando dal basso verso l'alto: si perdono i dettagli sull'implementazione, acquisendo una visione più semplificata e generale. Ciascun livello nasconde i dettagli e la complessità di quello sottostante

Livelli di astrazione

Algorithms
Programming Languages
Operating Systems
Instruction Set Architecture
Microarchitecture
Register Transfers
Logic Gates
Transistor Circuits

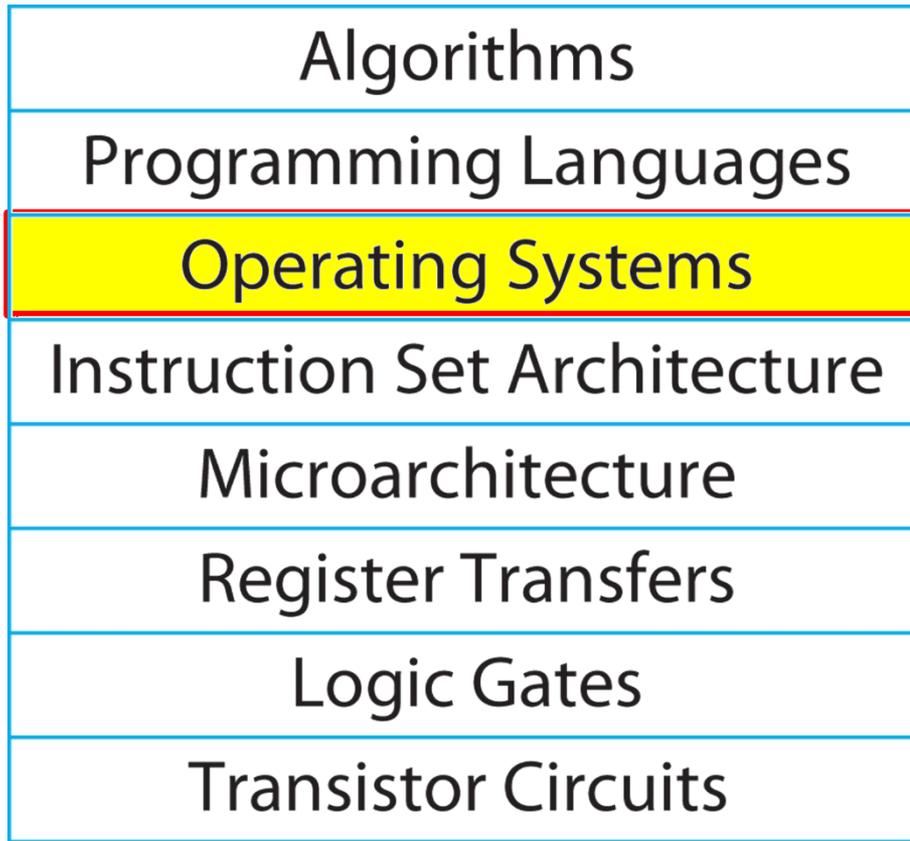
Algoritmo: descrizione di **sequenze di passi** per realizzare una certa operazione. L'algoritmo è **generico**, astrae da un particolare linguaggio

Livelli di astrazione

Algorithms
Programming Languages
Operating Systems
Instruction Set Architecture
Microarchitecture
Register Transfers
Logic Gates
Transistor Circuits

Linguaggio di programmazione (es. Java, C++): comprende una serie di **istruzioni per implementare l'algoritmo** e produrre certi dati in uscita. Può lavorare su diversi sistemi operativi

Livelli di astrazione



Sistema operativo:

controlla il flusso di esecuzione del programma, gestisce l'allocazione delle risorse tra i vari programmi. Può operare su diversi processori

Livelli di astrazione

Algorithms
Programming Languages
Operating Systems
Instruction Set Architecture
Microarchitecture
Register Transfers
Logic Gates
Transistor Circuits

ISA: insieme di istruzioni elementari nel linguaggio macchina (linguaggio del processore) in cui deve essere tradotto il programma per funzionare su un determinato processore

Livelli di astrazione

Algorithms
Programming Languages
Operating Systems
Instruction Set Architecture
Microarchitecture
Register Transfers
Logic Gates
Transistor Circuits

Microarchitettura:

specifica realizzazione (implementazione) del set di istruzioni (ISA) in un particolare processore

Livelli di astrazione

Algorithms
Programming Languages
Operating Systems
Instruction Set Architecture
Microarchitecture
Register Transfers
Logic Gates
Transistor Circuits

Trasferimento di dati tra registri: descrizione (es. tramite HDL) delle microoperazioni come sequenze di trasferimenti di dati, cioè gruppi di bit

Livelli di astrazione

Algorithms
Programming Languages
Operating Systems
Instruction Set Architecture
Microarchitecture
Register Transfers
Logic Gates
Transistor Circuits

Porte logiche:
mattoncini che descrivono le operazioni elementari da un punto di vista logico (non elettrico!). Le porte logiche eseguono una serie di **operazioni logiche sui bit**

Livelli di astrazione

Algorithms
Programming Languages
Operating Systems
Instruction Set Architecture
Microarchitecture
Register Transfers
Logic Gates
Transistor Circuits

Transistor: dispositivi elettronici che operano da interruttori. Sono i mattoncini che, opportunamente connessi tra loro, costituiscono le porte logiche. Un circuito a livello di transistor descrive il sistema da un punto di vista elettrico

Algorithms
Programming Languages
Operating Systems
Instruction Set Architecture
Microarchitecture
Register Transfers
Logic Gates
Transistor Circuits

Livelli di astrazione

- Il sistema può essere modificato ad un certo livello di astrazione, lasciando **invariati tutti i livelli sovrastanti**
 - Ciò permette di continuare ad usare soluzioni ad alti livelli di astrazione anche quando le implementazioni sottostanti sono cambiate
- E' importante **scegliere il giusto livello di astrazione** a cui lavorare
 - Trascurare i dettagli non strettamente necessari
 - Ottimizzare gli aspetti rilevanti per un certo tipo di design

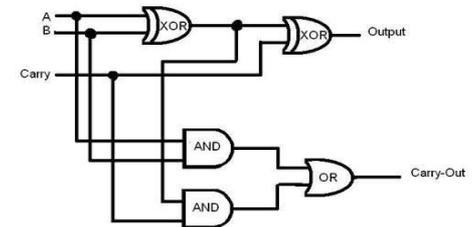
1)	Algorithms
2)	Programming Languages
3)	Operating Systems
4)	Instruction Set Architecture
5)	Microarchitecture
6)	Register Transfers
7)	Logic Gates
8)	Transistor Circuits

Esempio di livelli di astrazione: somma

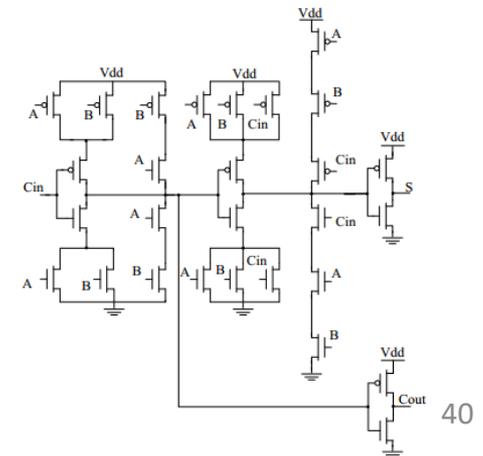
2) **Linguaggio di programmazione:** dice al sistema di eseguire l'addizione tra i valori contenuti in due variabili e immagazzinare il contenuto della somma in una terza variabile

$$S = a + b$$

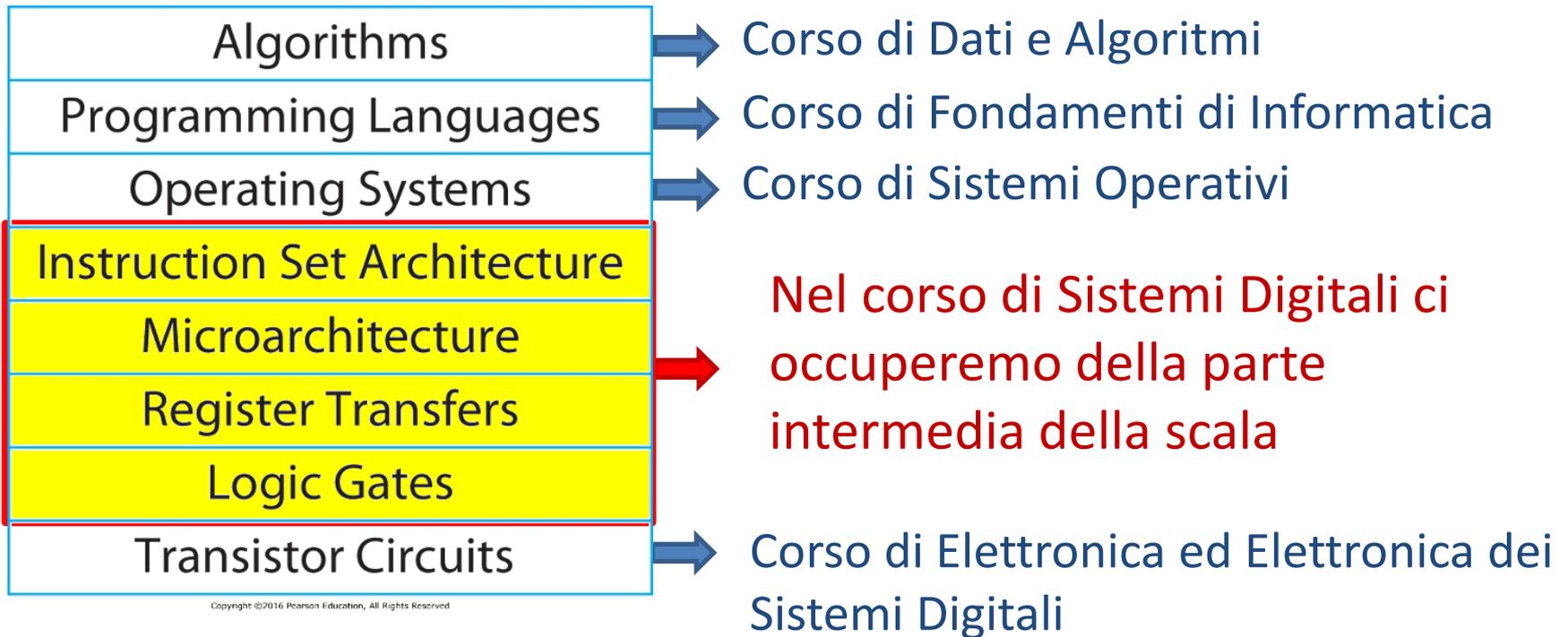
7) **Porte logiche:** dettaglio delle porte logiche del sommatore, i loro ingressi, uscite e connessioni



8) **Circuito a livello di transistor:** la somma viene effettivamente eseguita tramite un movimento di carica elettrica tra i nodi di un circuito, il cui risultato sarà rappresentare i bit della somma al nodo di uscita del circuito



Livelli di astrazione



Panoramica del processo di progettazione dei sistemi digitali e VHDL

Flusso di progettazione di un sistema digitale

- **Definizione delle specifiche:** comportamento del circuito (descrizione in forma di testo o codice HDL). Le specifiche astraggono dall'implementazione: rappresentano cosa fa il sistema, non come lo fa
- **Formulazione:** tabella di verità o equazioni booleane per definire la relazione logica tra ingressi e uscite che il sistema deve implementare
- **Ottimizzazione:** minimizzazione del numero di porte logiche (transistor, i.e. area di silicio). Vanno anche considerati vincoli come massimo numero di ingressi e uscite di una porta logica, ritardi, ecc. che dipendono dalla tecnologia. E' un processo iterativo, che spesso va ripetuto dopo la fase di mappatura
- **Mappatura tecnologica:** traduzione del diagramma logico/netlist in uno schema che contiene i componenti disponibili nella tecnologia scelta
- **Verifica:** controllo della correttezza del design

VHDL

- **VHDL** (Very high speed integrated circuits **H**ardware **D**escription Language) è uno dei linguaggi più usati per la progettazione di sistemi digitali
- *NON è un linguaggio di programmazione!*
 - In un linguaggio di programmazione, le istruzioni vengono eseguite in sequenza da una infrastruttura (es: CPU) => una sola istruzione alla volta è attiva (in un singolo core)
- Nel VHDL tutte le **dichiarazioni** sono attive allo stesso tempo, cioè vengono **eseguite in parallelo**
 - Gli statement del linguaggio definiscono blocchi di hardware
 - Non esiste un'infrastruttura sottostante, non c'è un'esecuzione sequenziale

VHDL: scopi di utilizzo

- **Documentazione** di un design
- **Simulazione** del comportamento di un design: studio dell'evoluzione dei segnali per valutare il corretto funzionamento del circuito
 - Applica una serie di segnali in ingresso al circuito e verifica le uscite
- **Sintesi automatica** di un design: passaggio da una descrizione comportamentale ad una descrizione a livello di porte logiche, i.e. traduzione del listato VHDL in una rete (netlist) di celle (logic cells) fisicamente realizzabili, che può essere a sua volta usata per produrre un circuito reale
 - Fornita da appositi programmi che si appoggiano a librerie dove sono descritte le porte logiche disponibili (fornite dal venditore)
- In questo corso NON ci occuperemo di sintesi di circuiti

VHDL: livelli di astrazione

- Potente e versatile, permette la descrizione di un circuito a **diversi livelli di astrazione**
 - **Structural**: descrive il sistema in forma di componenti interconnessi, è equivalente ad uno schematico → basso livello di astrazione
 - **Behavioral**: descrive il comportamento e la funzionalità del sistema, a prescindere dall'effettiva implementazione a livello di componenti → alto livello di astrazione
 - **Dataflow**: livello di astrazione intermedio tra structural e behavioral, descrive il sistema in forma di flusso di dati

Disclaimer

Figures from *Logic and Computer Design Fundamentals*,
Fifth Edition, GE Mano | Kime | Martin

© 2016 Pearson Education, Ltd