



La Sapienza

Università degli Studi di Roma

Dipartimento di Informatica e Sistemistica

CALCOLATORI ELETTRONICI

Logica digitale – parte 4

Emiliano Trevisani

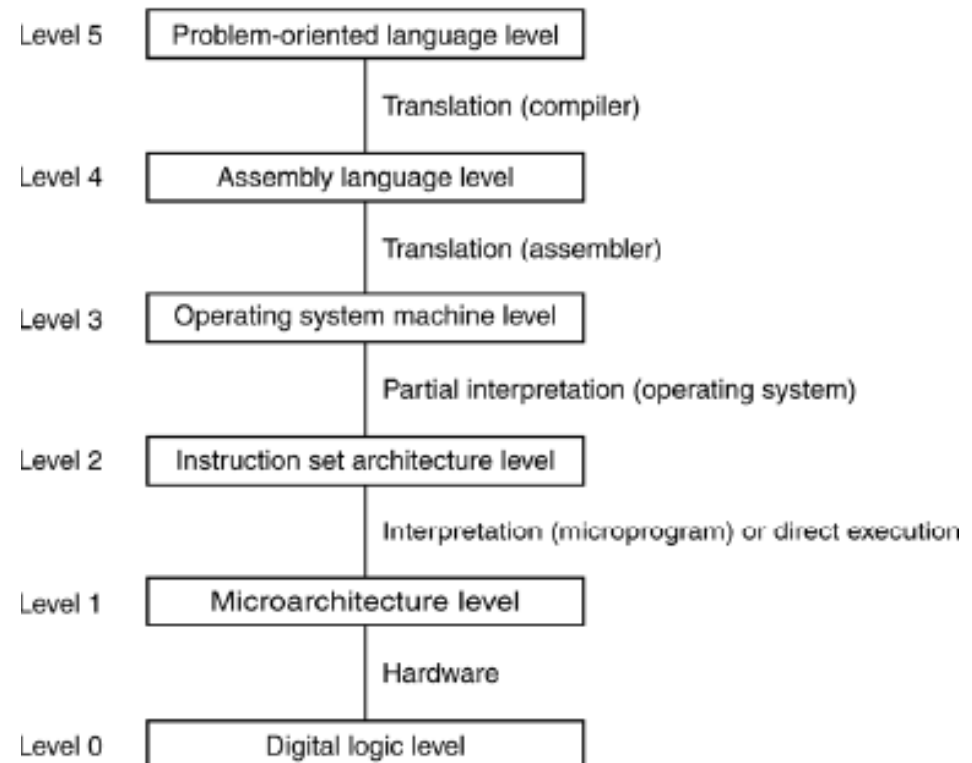
trevisani@dis.uniroma1.it

A.A. 2007/2008

Struttura di sistema di elaborazione



- ❑ L'approccio moderno al problema dell'"elaborazione automatica" è basato su un modello architetturale "a livelli" [layered architecture]
- ❑ Il livello 2 è il più basso al quale un utente può programmare la macchina
- ❑ Normalmente la programma a livello 5 [linguaggi di programmazione ad alto livello]
- ❑ Non tutti i sistemi hanno il livello 1



Struttura di sistema di elaborazione

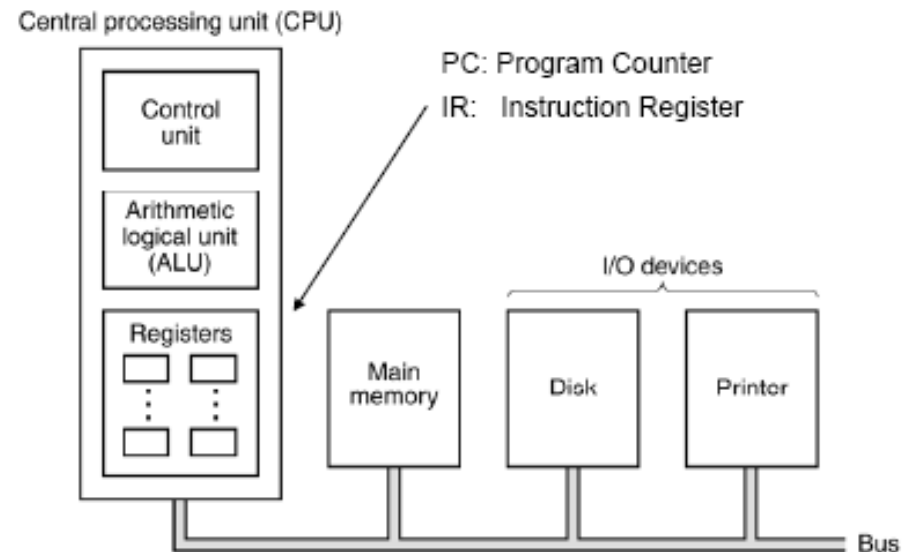


□ **Struttura del computer:**

- La memoria contiene sia i dati che le istruzioni
- Il contenuto dei registri può essere scambiato con la memoria e l'I/O
- Le istruzioni trasferiscono i dati e modificano il contenuto dei registri
 - Esempi: operazioni aritmetiche, lettura/scrittura da/verso memoria

▪ **Registri particolari:**

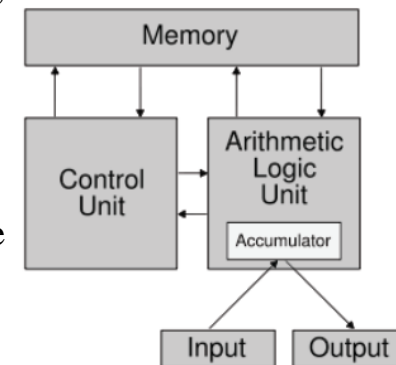
- **PC: [Program Counter indirizza l'istruzione corrente**
- **IR [Instruction Register contiene l'istruzione corrente**



Struttura di sistema di elaborazione



- I calcolatori moderni si ispirano ancora al modello astratto di macchina proposto nel 1944 ad Von Neumann
 - architettura dell'elaboratore suddivisa in blocchi logico-funzionali; programmi interpretati ed eseguiti **sequenzialmente**; programmi e dati nell'unità di memoria
 - 5 componenti fondamentali:
 1. CPU o unità di lavoro a sua volta suddivisa in
 - Unità operativa [il cui sottosistemi più rilevante è l'ALU]
 - Unità di controllo
 2. Unità di memoria, intesa come memoria di lavoro o memoria principale
 3. Unità di input, tramite la quale i dati vengono inseriti nel calcolatore per essere elaborati
 4. Unità di output, necessaria affinché i dati elaborati possano essere restituiti all'operatore
 5. Bus, ossia un canale per l'interconnessione dei componenti
 - All'interno dell'ALU è presente un registro detto accumulatore, agisce da buffer tra input e output grazie a una speciale istruzione che carica una parola dalla memoria all'accumulatore e viceversa.





□ Il ciclo istruzione [FETCH – DECODE - EXECUTE]:

- L'esecuzione di ciascuna istruzione nella CPU consta dei seguenti passi:
 1. Carica l'istruzione da memoria in IR (Instruction Register) (**Fetch**)
 2. Incrementa PC (Program Counter)
 3. Decodifica l'istruzione (**Decode**)
 4. Se l'istruzione usa un dato in memoria calcolane l'indirizzo
 5. Carica l'operando in un registro
 6. Esegui l'istruzione (**Execute**)
 7. Torna al passo 1 per l'esecuzione dell'istruzione successiva

Struttura di sistema di elaborazione



□ **Esecuzione ed Interpretazione delle istruzioni** [2 approcci]

▪ Esecuzione diretta

- Istruzioni eseguite direttamente dai circuiti hardware
- Approccio molto complesso:
 - Repertorio di istruzioni limitato
 - Progettazione dell'HW complessa
- Esecuzione molto efficiente

▪ Interpretazione

- L'hardware può eseguire solo alcune operazioni elementari molto semplici
- Ciascuna istruzione è scomposta in una successione di operazioni base fatte eseguire dall'hardware dalla logica di controllo
- Vantaggi:
 - Repertorio di istruzioni esteso
 - HW più compatto
 - Flessibilità di progetto

Struttura di sistema di elaborazione



□ **Microprogrammazione**

- L'HW può eseguire microoperazioni:
 - Trasferimenti tra registri
 - Trasferimenti da e per la memoria
 - Operazioni della ALU su registri
- Ciascuna istruzione viene scomposta in una sequenza di microoperazioni
- L'unità di controllo della CPU [**rete sequenziale**] esegue un microprogramma per effettuare l'interpretazione delle istruzioni
- Il microprogramma è contenuto in una memoria ROM sul chip del processore (Firmware)
- Vantaggi:
 - Disegno strutturato
 - Semplice correggere errori
 - Facile aggiungere nuove istruzioni [architettura estensibile]



□ Architetture CISC e RISC

- All'inizio degli anni '80 i progettisti di sistemi veloci riconsiderarono l'approccio **esecuzione diretta**
 - Obiettivi di progetto architetture RISC:
 - **Eseguire direttamente** le istruzioni più frequenti
 - Massimizzare la frequenza alla quale le istruzioni sono eseguite [misurata in MIPS (Millions of Instructions Per Second)]
 - Semplificare la decodifica delle istruzioni: formati molto regolari
 - Limitare i riferimenti alla memoria alle sole LOAD e STORE
 - Ampliare il numero di registri per limitare l'uso di LOAD e STORE
 - Questi principi sono tipici della filosofia RISC ma anche le moderne architetture CISC (tipo Intel) si ispirano in parte ad essi

Struttura di sistema di elaborazione



□ Architetture CISC e RISC

- Architetture **RISC** (*Reduced Instruction Set Computer*):
 - Repertorio di istruzioni ristretto (alcune decine)
 - Istruzioni prevalentemente su registri pipeline
 - Programmi complessi: più istruzioni semplici per realizzare una funzione più complessa
 - La predicibilità delle poche operazioni implementate permette l'utilizzo di pipeline molto lunghe
 - **Una istruzione per ciclo di clock**
- Architetture **CISC** (Complex Instruction Set Computer) **μ-programmate**:
 - Repertorio di istruzioni esteso (alcune centinaia)
 - Istruzioni anche su memoria
 - Programmi più semplici (meno istruzioni per programma)
 - Ottimizzare istruzioni molto complesse rende difficile l'utilizzo estensivo delle tecniche di pipelining
 - **Molti cicli di clock per istruzione**

Struttura di sistema di elaborazione



□ Architetture CISC e RISC

- Esempi:
 - Alpha(DEC) e IBM RS6000: RISC
 - PentiumII (Intel): CISC
- Nonostante gli evidenti vantaggi delle architetture RISC queste non hanno sostituito quelle CISC a causa dei problemi di compatibilità con i modelli già esistenti e dell'enorme mole di sviluppi hw e sw già sostenuti
- Attualmente le differenze tra le 2 architetture sono troppo sfumate per poter classificare una CPU secondo l'uno o l'altro modello

Struttura di sistema di elaborazione



□ **Parallelismo di calcolo**

- Il parallelismo è ormai l'unica strada per aumentare le prestazioni
- Limite di un'esecuzione sequenziale: velocità della luce (30 cm in 1 ns)
- Due tipi di parallelismo:
 - A livello di istruzioni (Pipelining)
 - diverse istruzioni eseguite insieme
 - diverse fasi della stessa istruzione eseguite in concorrenza
 - A livello di processori
 - più processori lavorano insieme allo stesso problema
 - fattori di parallelismo molto elevati
 - diversi tipi di interconnessione e di cooperazione
 - Reti di calcolatori e cluster: I sistemi lavorano in parallelo con diverse specializzazioni
- Aumentare eccessivamente il grado di parallelismo determina problemi di gestione dei conflitti per i troppi accessi simultanei a risorse condivise [es. registri,...]



❑ Pipelining

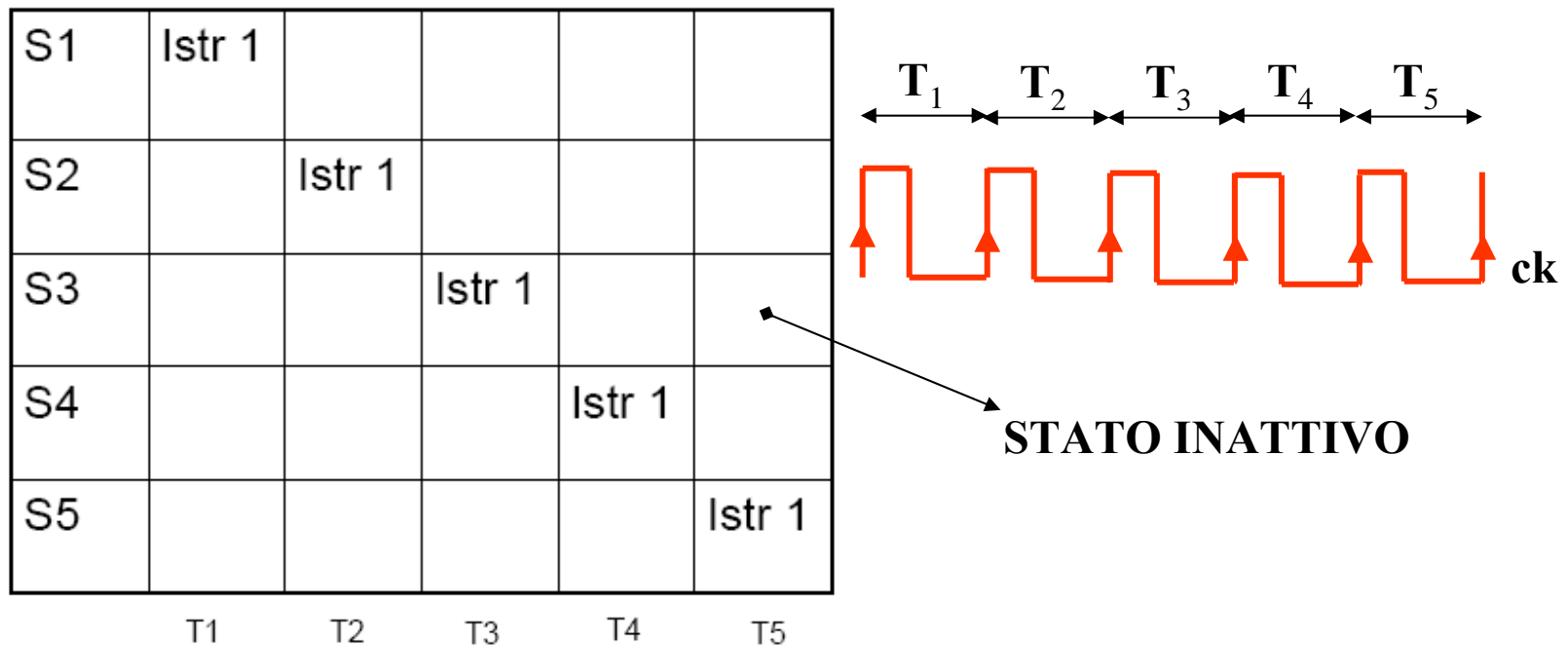
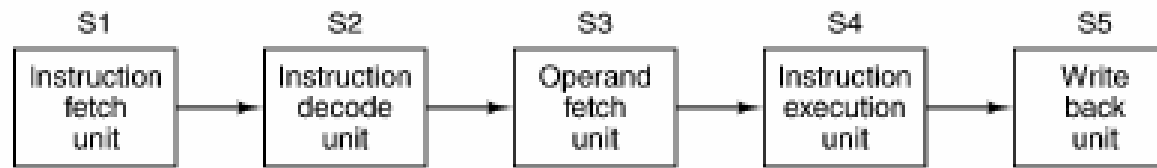
- L'esecuzione sequenziale delle istruzioni una per volta fino al completamento prevede l'esecuzione sequenziale dei corrispondenti cicli istruzioni [Fetch – Decode - Execute]
- In questo modo, restano inattivi (fase per fase) alcuni circuiti del sistema di calcolo non interessati dalla fase corrente del ciclo
- Idea base delle tecniche di pipelining: aumentare la velocità complessiva del sistema facendo in modo che il microprocessore esegua simultaneamente **fasi** diverse di **istruzioni** diverse
 - In questo modo, si tende a non lasciare inattivo alcun sottosistema della CPU
 - Istruzioni successive sono portate in esecuzione sequenzialmente senza necessariamente aspettare il completamento dell'istruzioni precedente
 - L'efficacia di una pipeline è legata al numero dei suoi stadi (lunghezza della pipeline); esiste tuttavia un limite oltre il quale non è conveniente estendere la pipeline: maggiore è la sua lunghezza maggiore è infatti la probabilità di entrare in stallo dovendo inserire "bolle" che fanno crollare le prestazioni.

Struttura di sistema di elaborazione



❑ Pipelining

- ❑ Esempio: Una istruzione, 5 Stati, 5 impulsi di clock. No Pipelining.
 - ogni impulso di clock attiva un solo stato

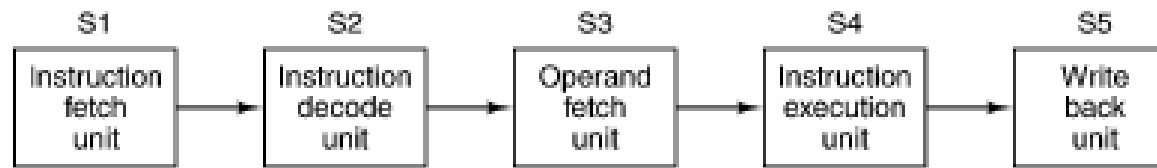


Struttura di sistema di elaborazione

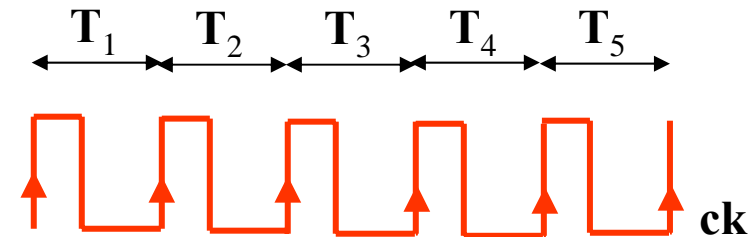


❑ Pipelining

- ❑ Esempio: Una istruzione, 5 Stati, 5 impulsi di clock. Pipelining a 2 livelli
 - ogni impulso di clock attiva 2 stati



S1	Istr 1	Istr 2				
S2		Istr 1	Istr 2			
S3			Istr 1	Istr 2		
S4				Istr 1	Istr 2	
S5					Istr 1	Istr 2
	T1	T2	T3	T4	T5	T6



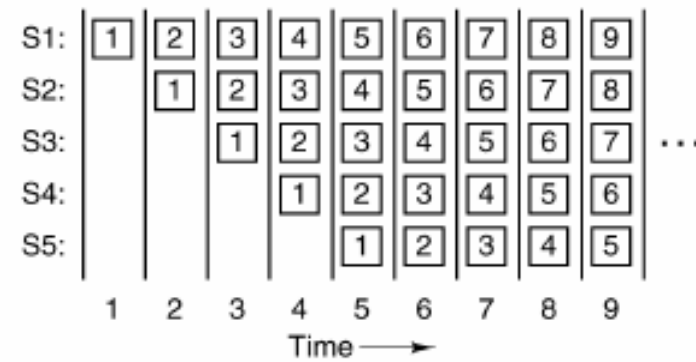
**Introdotta parallelismo ⇒
STATI INATTIVI
diminuiscono**

Struttura di sistema di elaborazione



□ Pipelining

- Esempio: Una istruzione, 5 Stati, 5 impulsi di clock. Pipelining
 - T_1 : lo stadio 1 lavora alla istruzione 1 e la legge dalla memoria
 - T_2 : lo stadio 2 decodifica l'istruzione 1 mentre lo stadio 1 legge l'istruzione 2
 - T_3 : lo stadio 3 legge gli operandi per l'istruzione 1, lo stadio 2 decodifica l'istruzione 2 e lo stadio 1 legge la terza istruzione
 -
- Ciascuna istruzione è divisa in fasi; la sua esecuzione avviene in una pipeline a più stadi
- **Più istruzioni in esecuzione simultanea; una istruzione completata per ogni ciclo**

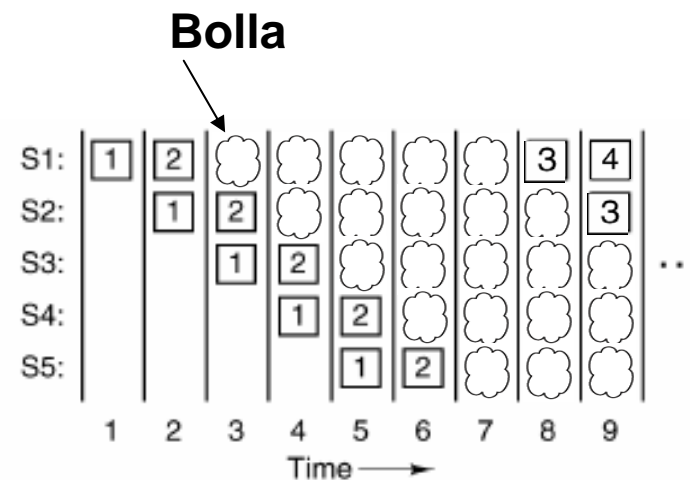


Struttura di sistema di elaborazione



□ Pipelining

- Problema: **compatibilità dell'esecuzione parallela**
 - Indipendenza tra le istruzioni; ciascuna istruzione non deve utilizzare i risultati dell'altra
 - La presenza di istruzioni “incompatibili” tra loro può portare la pipeline ad entrare in **stallo** per un certo numero di cicli di clock [inserimento di **bolle**]
 - Esempio: istruzione 3 utilizza il risultato dell'istruzione 2





□ Pipelining

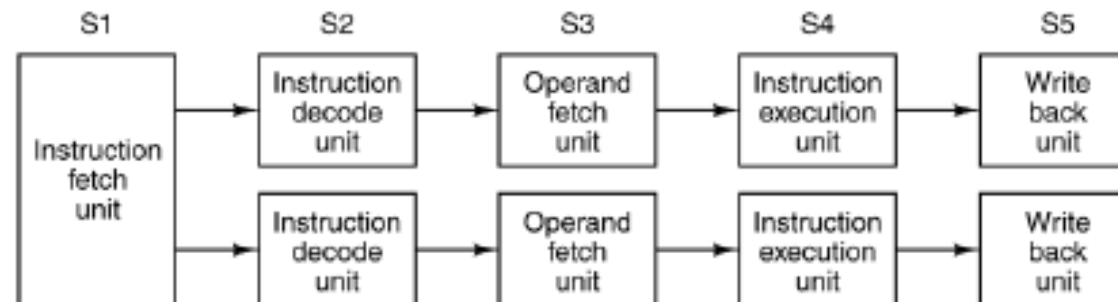
- Problema: **compatibilità dell'esecuzione parallela**
 - Riduzione della probabilità di stallo \Rightarrow diversi approcci:
 - Intervenire a livello di instruction set per fare in modo che le istruzioni risultino più “compatibili” (più facile per CPU RISC)
 - Intervenire a livello di programma riducendo la dipendenza tra operazioni successive
 - Intervenire a livello di programma in esecuzione riordinandone dinamicamente il codice
 - In generale i processori RISC riescono a gestire e mantenere “piene” pipelines composte da alcune decine di stadi [ridotto set di istruzioni]
 - I processori CISC, utilizzando un set istruzioni più esteso e variegato rendono intrinsecamente complessa l'implementazione di pipeline “lunghe”.

Struttura di sistema di elaborazione



□ Architetture superscalari

- Si aumenta il parallelismo introducendo più di una pipeline nel microprocessore
- Le pipeline possono essere specializzate:
 - Il Pentium ha 2 pipeline a 5 stadi; una può eseguire solo istruzioni su interi

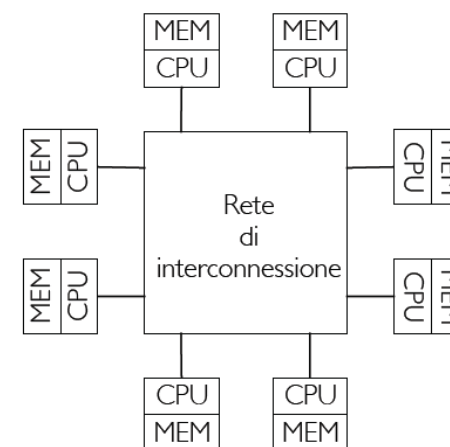
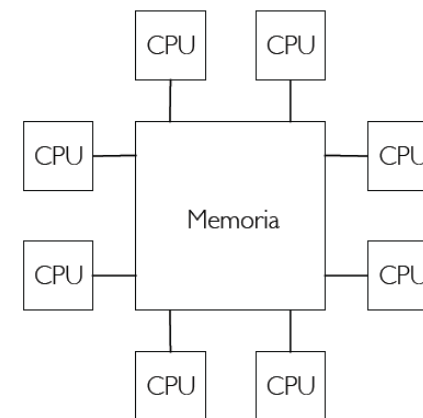


Struttura di sistema di elaborazione



□ Parallelismo di calcolo a livello processori

- Calcolatori con più processori
 - Architetture multiprocessore / multi-core
 - Risorse condivise [es. memoria]
 - Approccio costoso
- Multicomputer
 - Cooperazione fra sistemi di calcolo indipendenti
 - Necessaria rete di interconnessione
 - Risorse di calcolo indivise
 - Approccio meno costoso



Sistemi di memoria



- ❑ I sistemi di memoria di un elaboratore possono essere così classificati:
 - Memoria interna al processore
 - Memoria principale
 - Memoria secondaria
- ❑ **Memoria interna** [es: registri interni della CPU, memorie cache]
 - Registri visibili o no al programmatore
 - Memorizzano temporaneamente dati e istruzioni (vedi interconnessione registri con circuiti di calcolo)
 - Dimensioni dei registri: decine di bytes; tempo di accesso: ordine dei ns; costo elevato. Tecnologie utilizzate: tipicamente FF, SRAM
 - Nelle CPU più recenti cresce la quantità di risorse dedicate alla memoria. Infatti, con riferimento alle memorie cache nella CPU:
 - 1980: processori senza cache (i386)
 - 1995: Alpha 21164 55% dei transistors
 - 2000: Merced (Intel-HP) 85% dei transistors

Sistemi di memoria



□ Memoria principale [memoria di lavoro]

- E' la memoria nella quale sono contenuti i programmi che la CPU esegue e i dati che la stessa CPU può accedere **direttamente** attraverso il meccanismo delle gerarchie di memoria
- Dimensioni: centinaia di Mbytes su un personal computer, nell'ordine dei Gigabytes su supercalcolatori.
- Velocità: ordine dei 100 ns; tecnologie utilizzate: tipicamente DRAM

□ Memoria secondaria [memoria di massa]

- Dimensioni: ordine dei Gbytes/Therabytes.
- Velocità: ordine dei **milioni di ns** (millisecondi)
- Tecnologie utilizzate: dischi magnetici / ottici



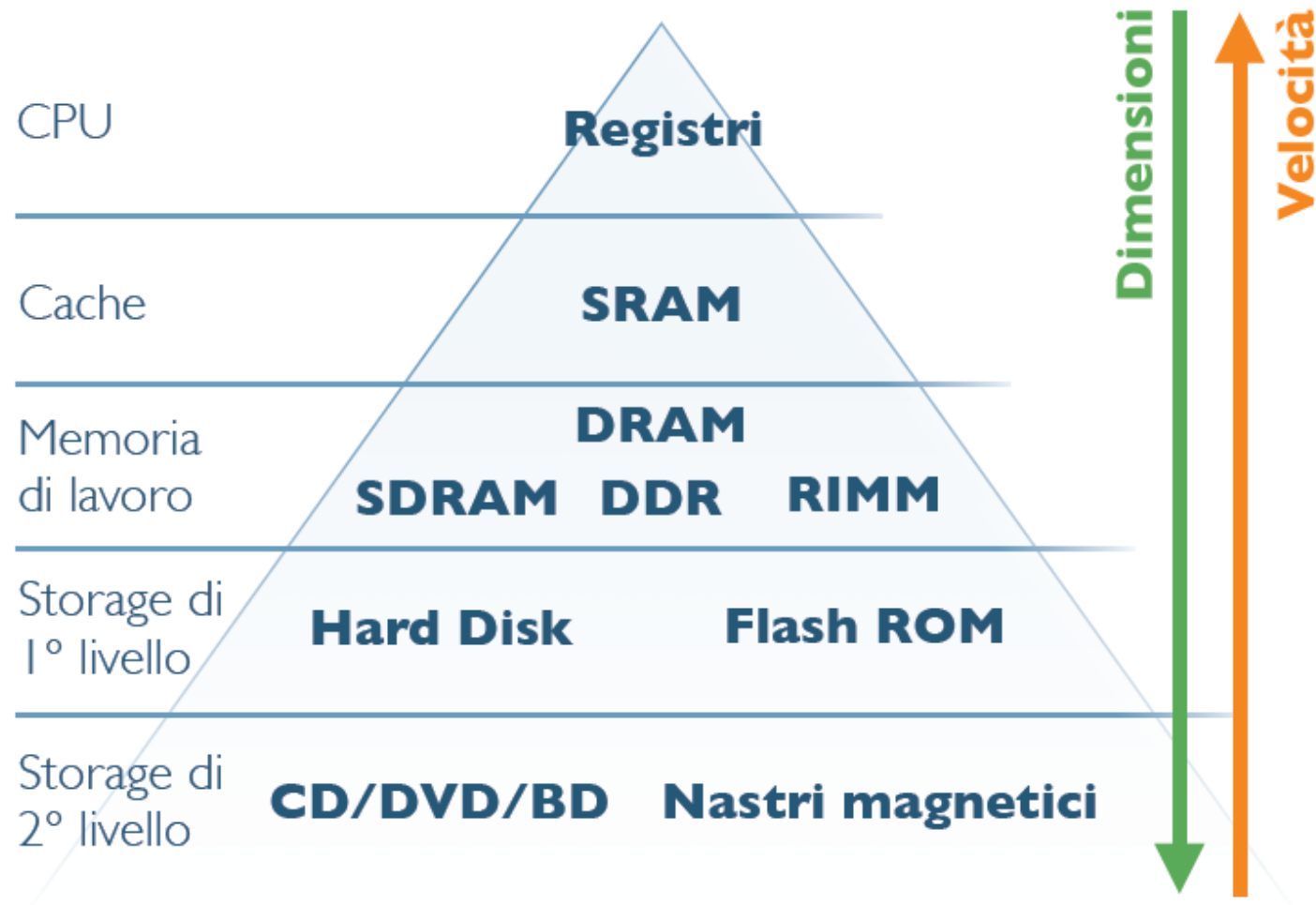
□ Tecnologie e caratteristiche delle memorie

- La singola tecnologia realizzativa è caratterizzata da:
 - costo per singolo bit di memoria
 - tempo di accesso (ritardo tra l'istante di richiesta del dato memorizzato e l'istante nel quale il dato è disponibile)
 - modo di accesso (seriale o casuale)
- Tecnologie:
 - Memorie a semiconduttore con tecnologia VLSI (memoria interna e principale)
 - Memorie magnetiche (memoria secondaria).
 - Memorie ottiche (memoria secondaria).

Sistemi di memoria



□ Gerarchia di memoria

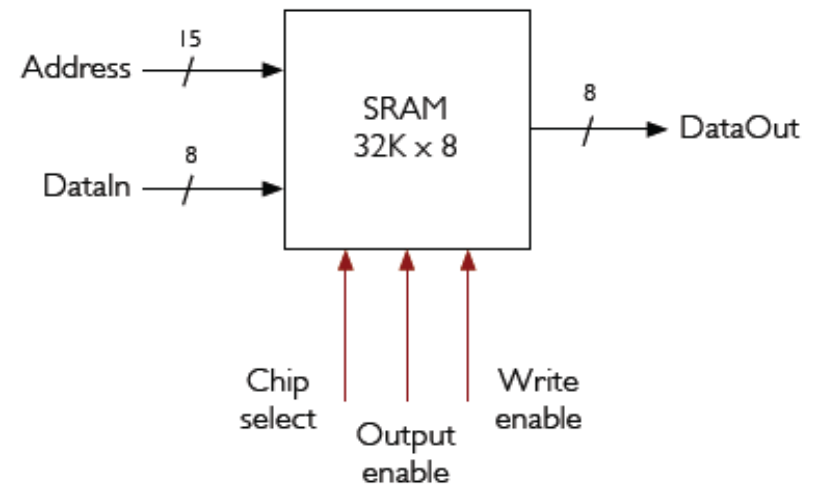


Sistemi di memoria



❑ SRAM: Static Random Access Memory

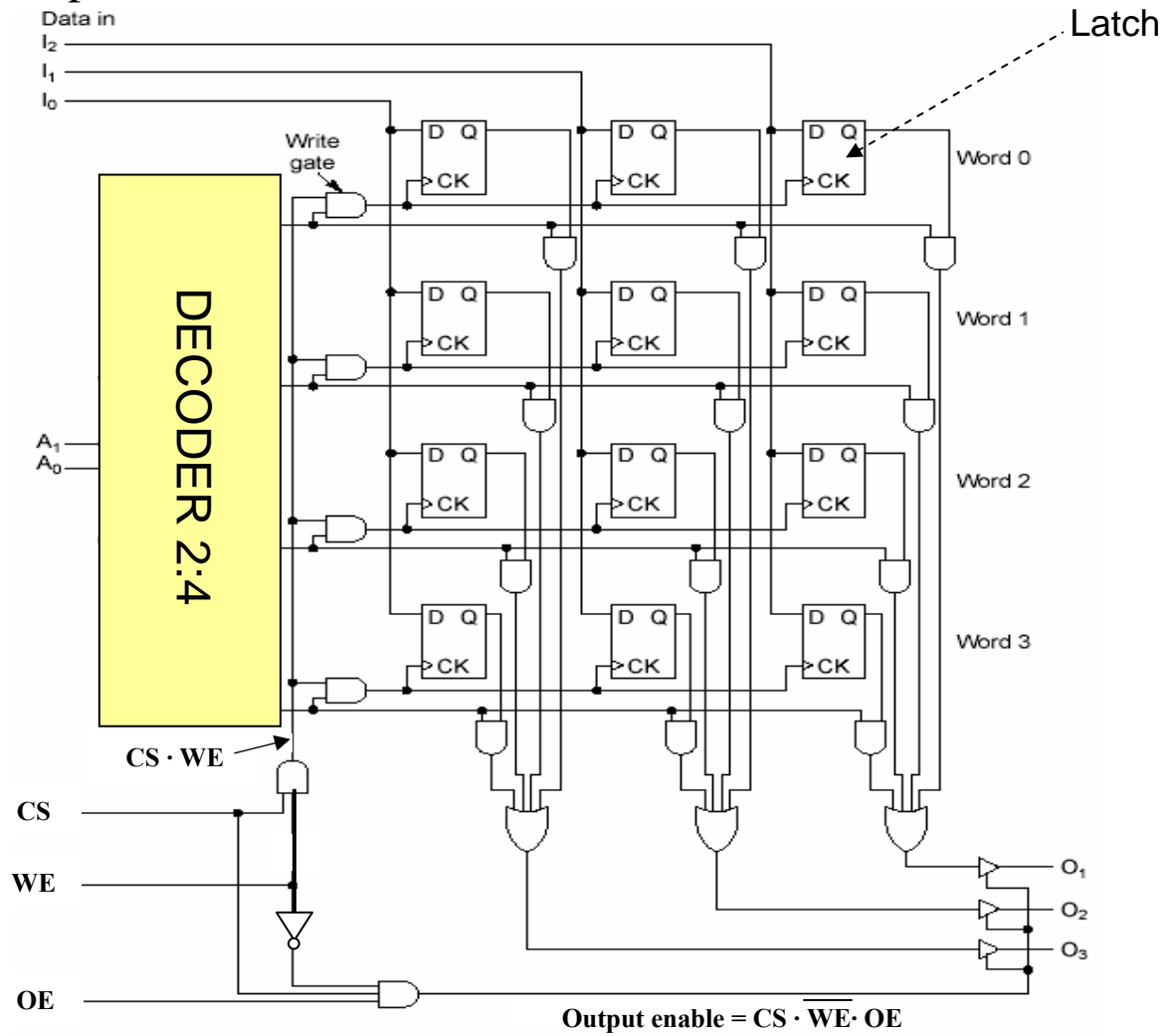
- La cella elementare di memoria è costituita da 4-6 transistor MOS che realizzano un FLIP-FLOP.
- L'informazione permane stabile in presenza della tensione di alimentazione
- Veloce, costosa, bassa densità (bit/area): tempo accesso 0.5 ÷ 5 ns, costo €4.000 ÷ €10.000 per GB (nel 2004); sebbene velocissime, sono costose e troppo poco “dense”
- usate nelle memorie cache (integrate nella CPU)
- chip SRAM caratterizzato da **altezza x larghezza del chip**
 - numero di locazioni di memoria indirizzabili (altezza del chip)
 - numero di bit memorizzati in ogni locazione (larghezza del chip)
 - Esempio: 32K x 8 SRAM



Sistemi di memoria



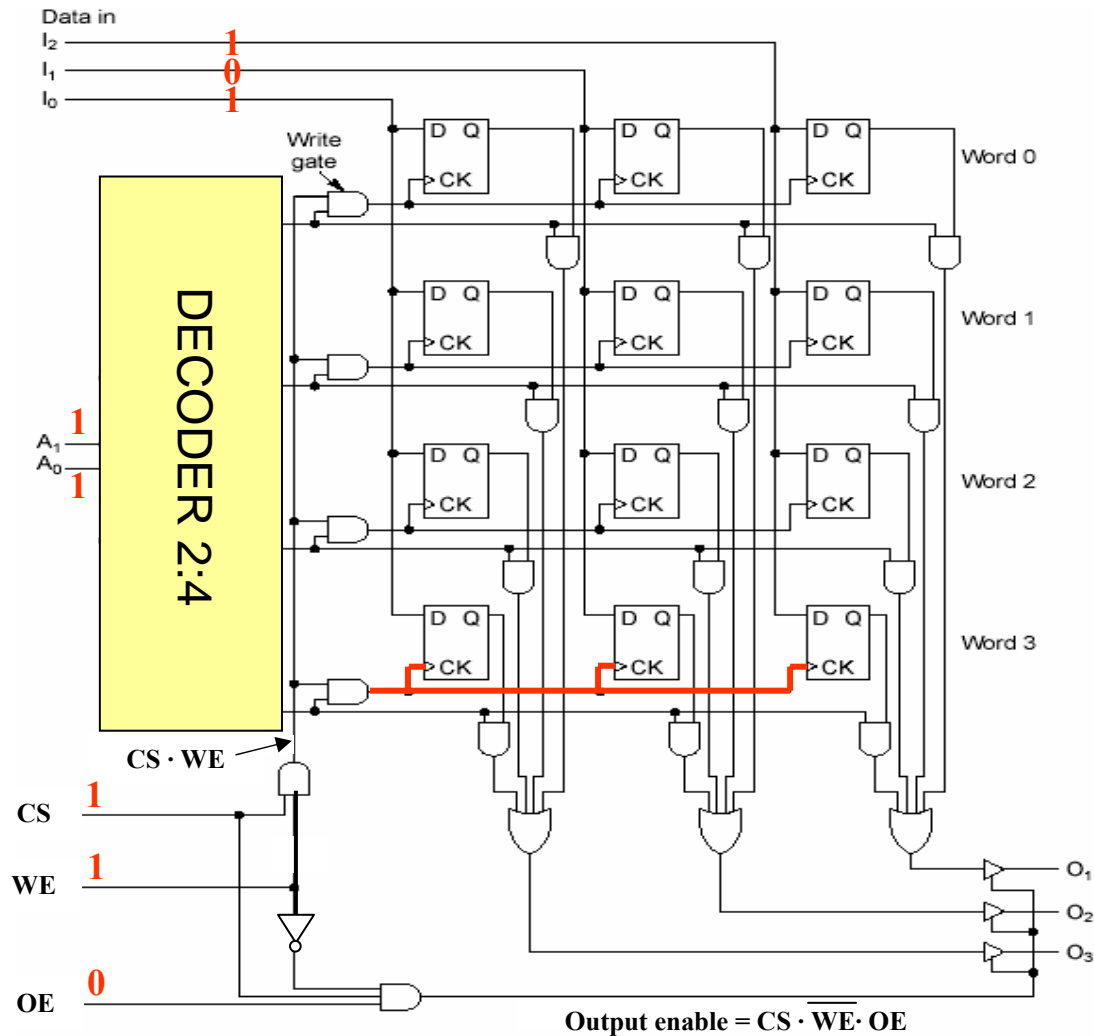
□ SRAM: Esempio di struttura interna SRAM 4 x 3



Sistemi di memoria



- **SRAM:** Esempio: SRAM 4 x 3 – scrittura in memoria del dato 101 all'indirizzo 11

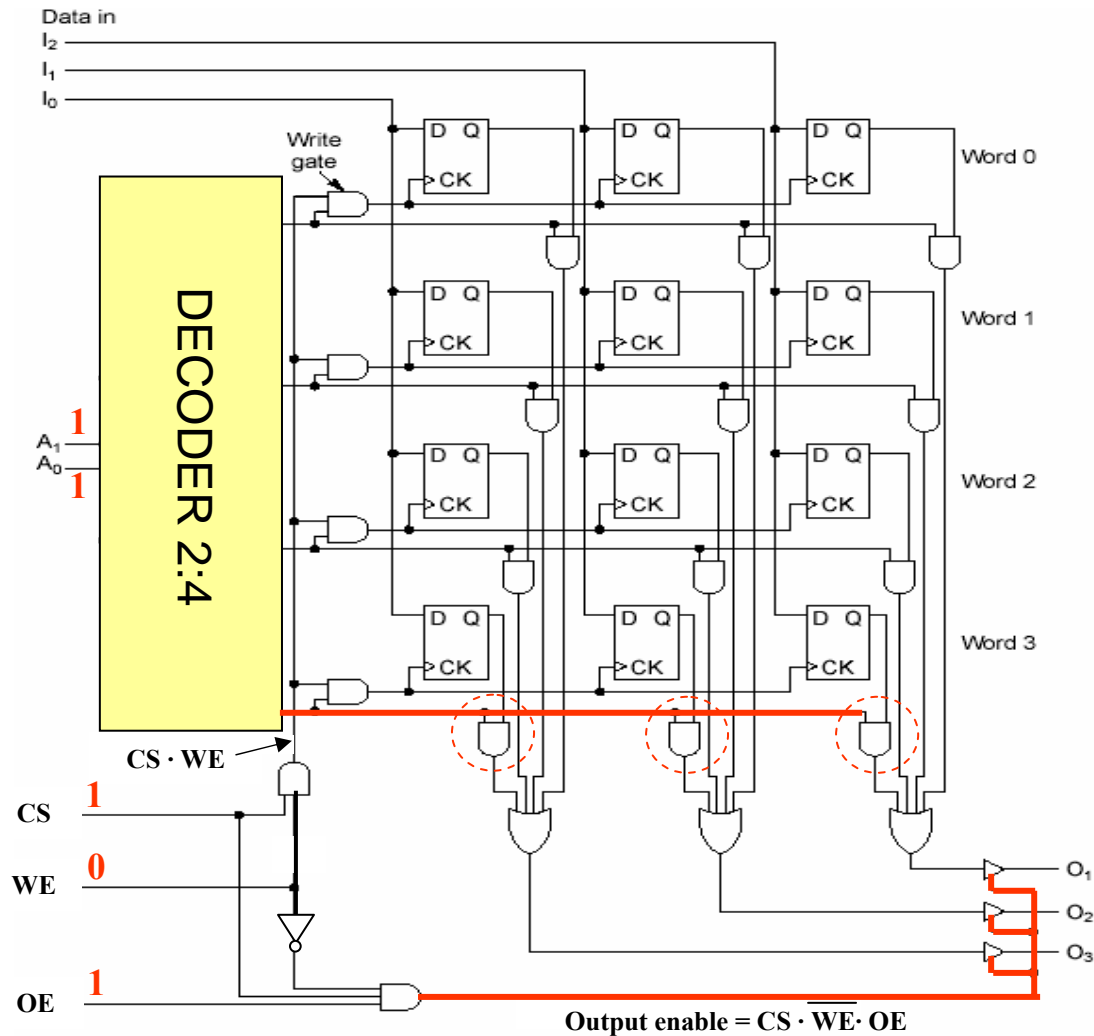


Sistemi di memoria



□ **SRAM:** Esempio: SRAM 4 x 3 – **lettura** in memoria del dato all'indirizzo

11

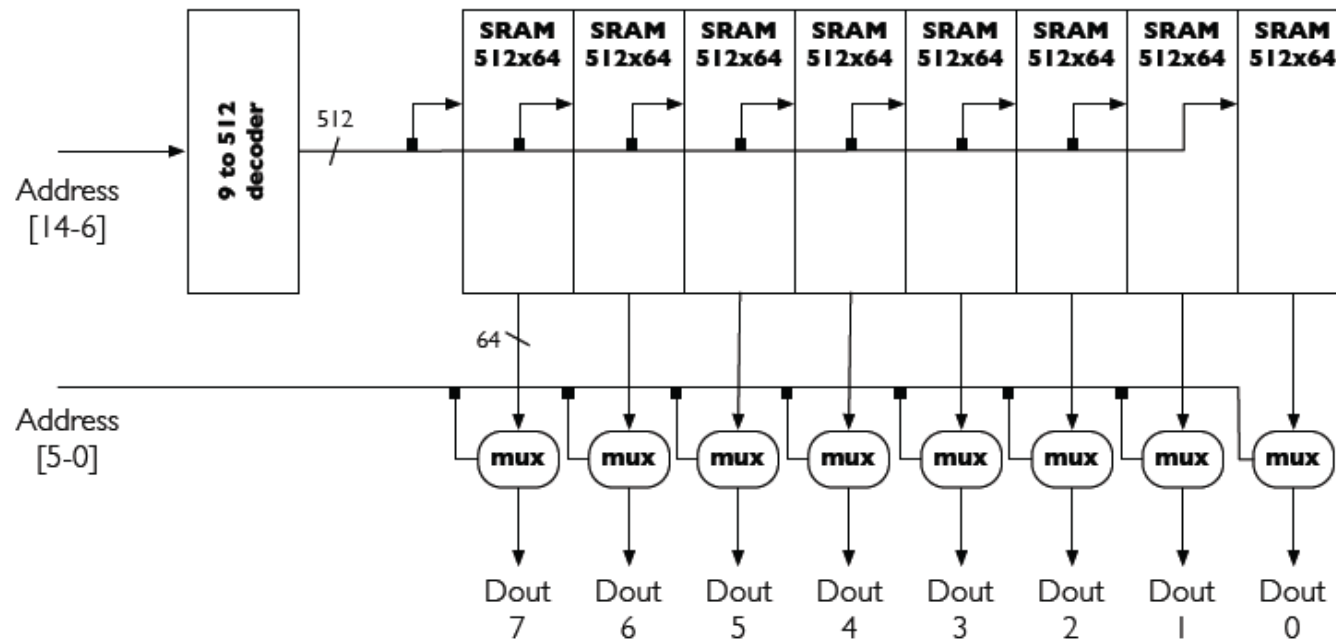


Sistemi di memoria



□ SRAM: Decodifica a 2 livelli

- Approccio utilizzato per evitare decoder d'ingresso ad elevato numero di ingressi / uscite
- Esempio: Ipotizzando di avere a disposizione un decoder 9 a 512 mostrare lo schema implementativo di SRAM 32K x 8
 - **Decoder + batteria di MUX; il decoder fissa lo schema implementativo**
 - Oss: senza quest'approccio, il decoder d'ingresso avrebbe avuto 2^{15} [32K] uscite
 - Oss: fissati i 9 bit più significativi di Address[bit 14-6] i restanti 6 bit [5-0] indirizzano 64 parole da 8 bit che sono memorizzate in modo "allineato"

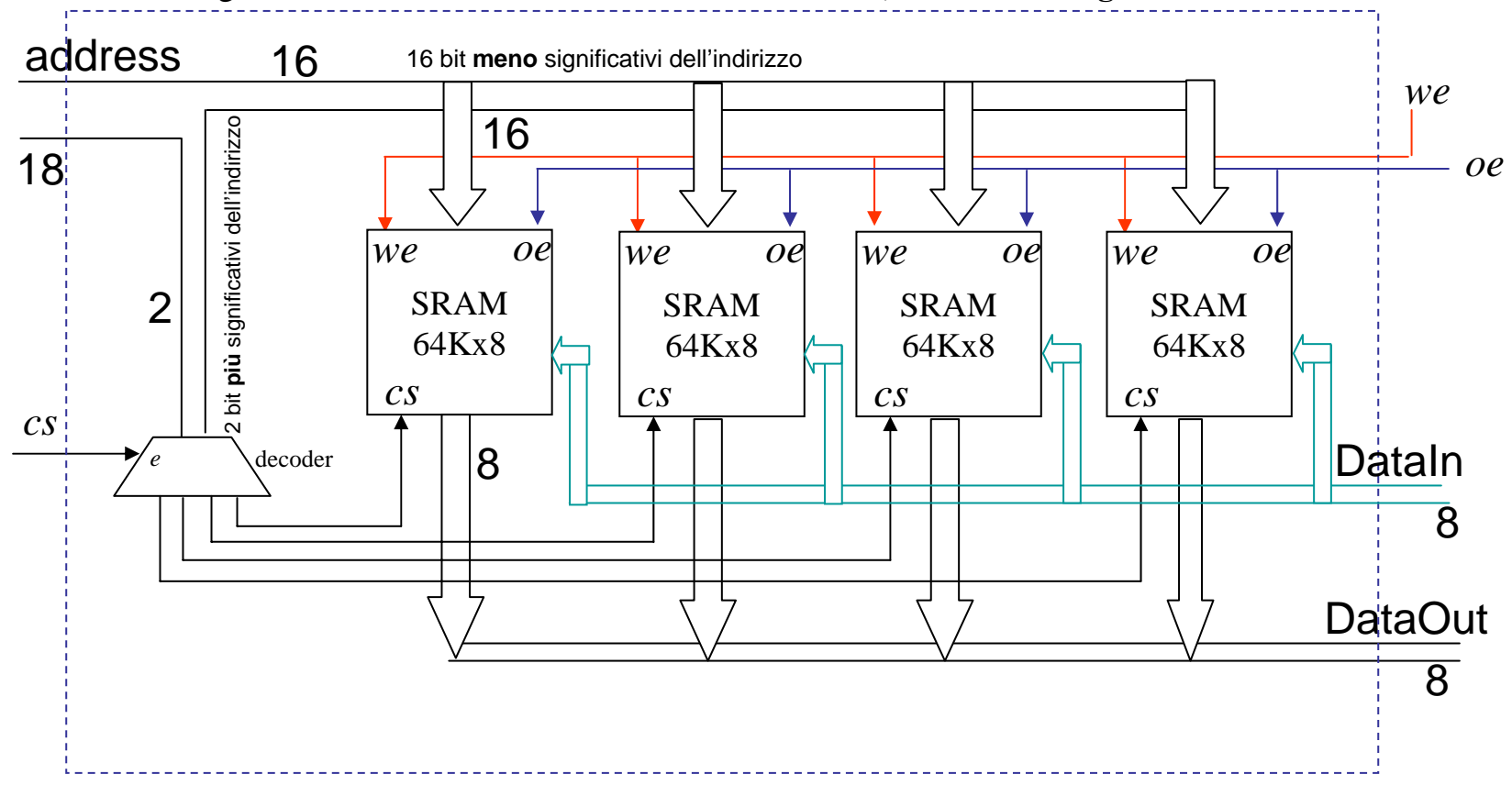


Sistemi di memoria



□ SRAM: Ampliamento [Più parole della stessa lunghezza]

- Esempio: Ipotizzando di avere a disposizione banchi SRAM 64Kx8 e circuiti combinatori di base, mostrare lo schema implementativo di una SRAM 256K x 8
 - Approccio 1 [analogia con esempio ampliamento ROM]: parole SRAM successive in genere nello stesso banco [**indirizzo di modulo** | **indirizzo di riga**]

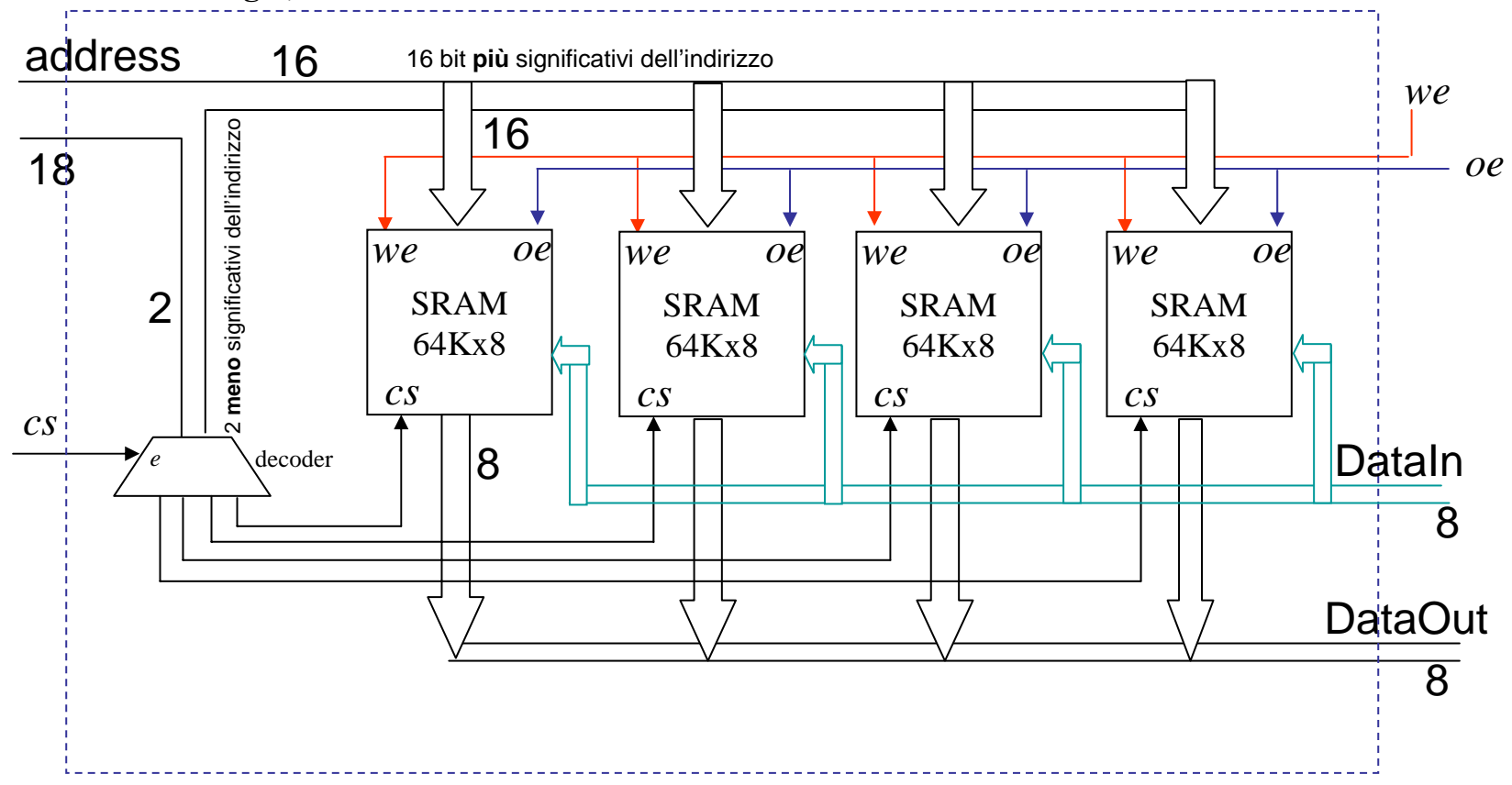


Sistemi di memoria



□ SRAM: Ampliamento [Più parole della stessa lunghezza]

- Esempio: Ipotizzando di avere a disposizione banchi SRAM 64Kx8 e circuiti combinatori di base, mostrare lo schema implementativo di una SRAM 256K x 8
 - Approccio 2 : parole SRAM successive in genere su banchi adiacenti [**indirizzo di riga** | **indirizzo di modulo**]

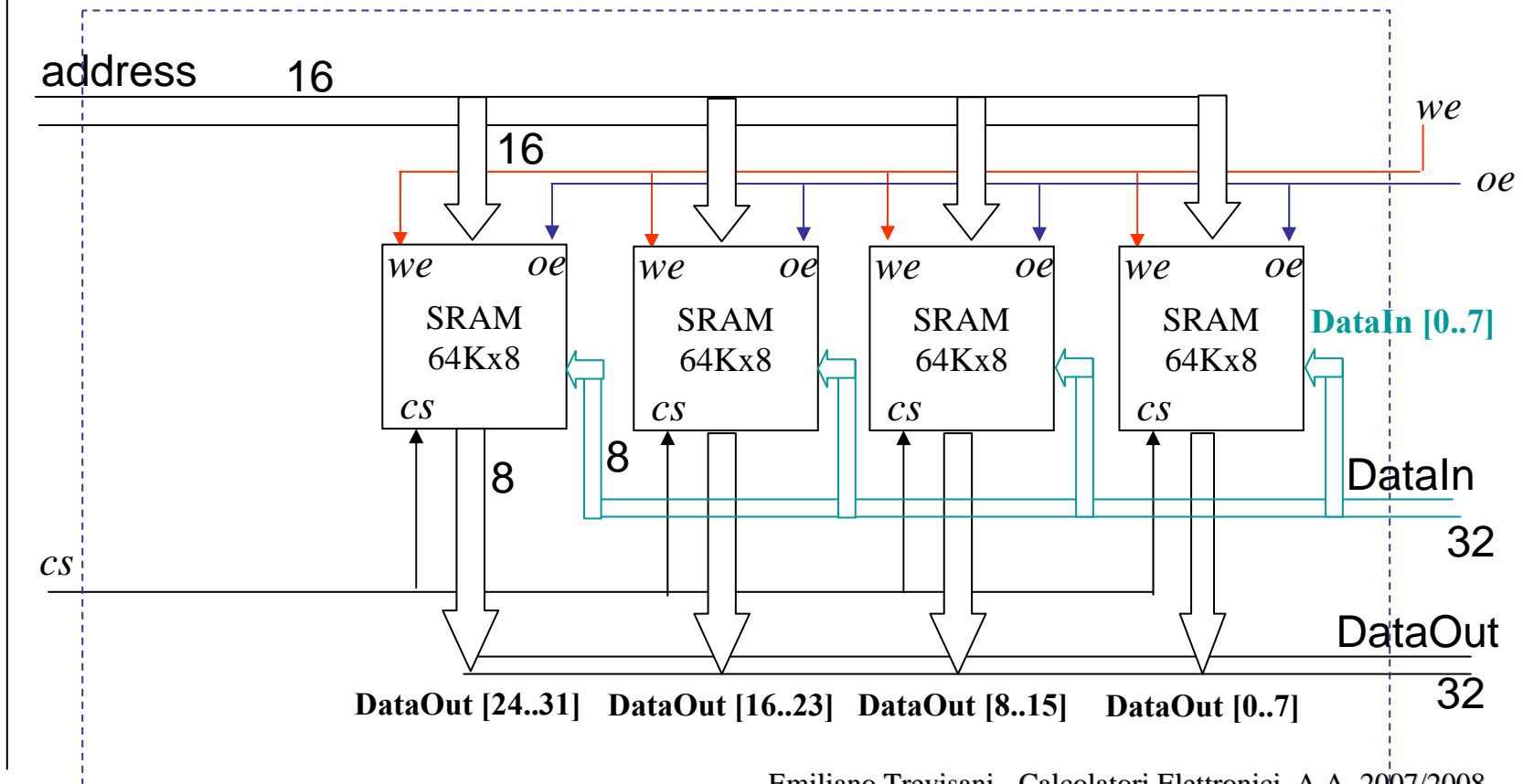


Sistemi di memoria



□ SRAM: Ampliamento [Stesso numero di parole ma più lunghe]

- Esempio: Ipotizzando di avere a disposizione banchi SRAM 64Kx8 e circuiti combinatori di base, mostrare lo schema implementativo di una SRAM 64K x 32

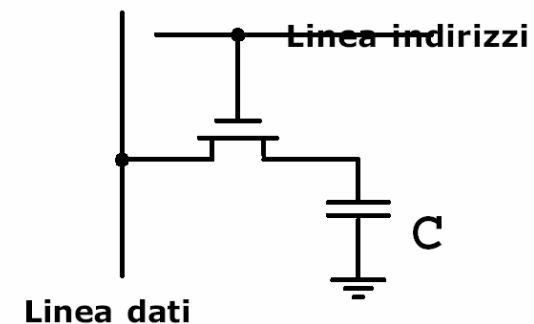


Sistemi di memoria



□ DRAM: Dynamic Random Access Memory

- La cella elementare è costituita da un condensatore che viene caricato (1) o scaricato (0) da un transistor di controllo
- La tensione sul condensatore tende a diminuire (millisecondi) e quindi deve essere ripristinata (refresh)
 - Cicli periodici di lettura / scrittura
 - Accesso lento [20 ÷ 100 ns]
- La semplicità della cella consente capacità molto elevate in spazi (e costi) contenuti
- Vantaggi: alta densità, basso costo [100 €/per GB nel 2004]
- Synchronous DRAM (SDRAM): lettura sincrona ad un segnale di clock (un solo fronte attivo)
 - Sincrona, comandata dal clock principale e non dai segnali di controllo, velocizza lo scambio dati tra memoria e CPU
 - DDR SDRAM (Double Data Rate)
 - 2 fronti del clock attivi
 - DDR2 SDRAM
 - DDR3 SDRAM (coming soon...)



Sistemi di memoria



□ **DRAM:** Dynamic Random Access Memory

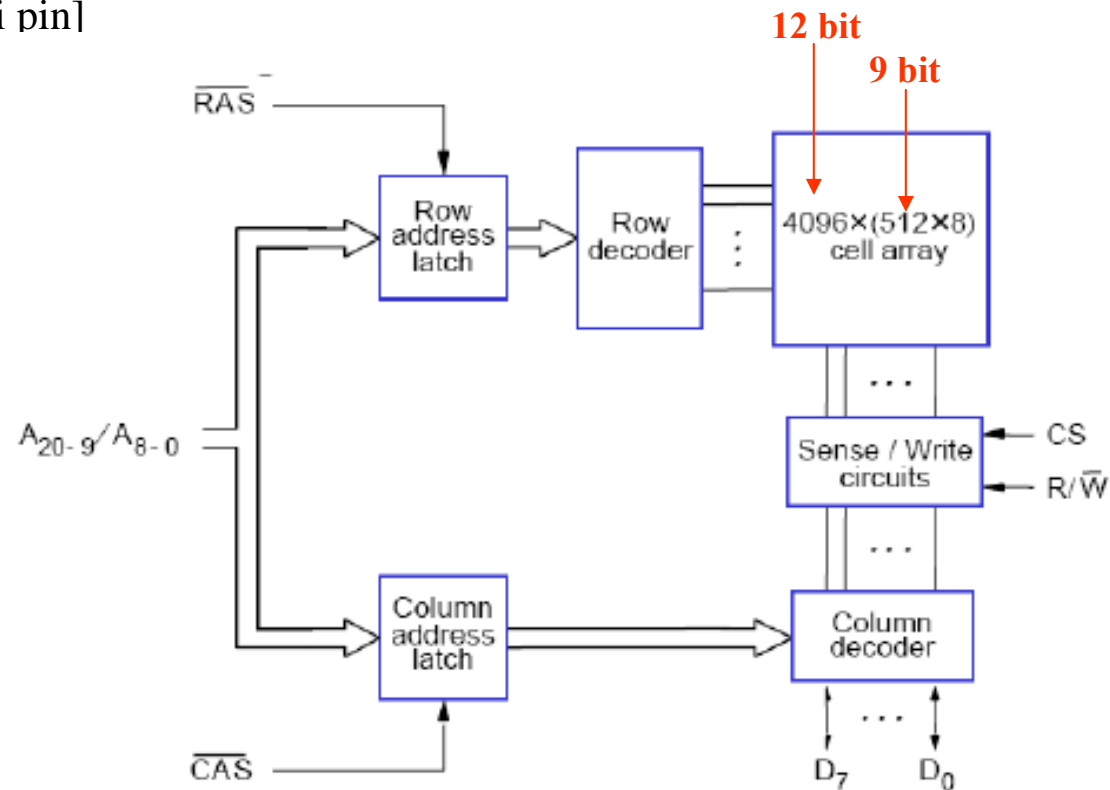
- Essendo memorie tipicamente ad alta capacità, si utilizza un indirizzamento a 2 livelli [riga / colonna]
 - Decoder a due livelli per l'indirizzamento
 - L'indirizzo di riga viene prelevato dal bus indirizzi e **memorizzato** dal memory controller in una sorta di registro (Row Address Latch) utilizzando il segnale di controllo RAS [Row Address Strobe]
 - L'indirizzo di colonna viene prelevato dal bus indirizzi e **memorizzato** dal memory controller in una sorta di registro (Column Address Latch) utilizzando il segnale di controllo CAS [Column Address Strobe]
 - Gli indirizzi di riga e colonna sono decodificati ed individuano una locazione: il dato memorizzato è disponibile sul bus dati
 - Oss: successive operazioni di lettura sulla stessa riga richiedono una modifica solo dell'indirizzo di colonna [CAS] (più efficiente e veloce)

Sistemi di memoria



□ DRAM: Esempio 2Mx8

- Indirizzamento **logico** a 21 bit [$2M=2^{21}$]
- Bus indirizzi a **12** linee: gli indirizzi di riga e colonna sono specificati in tempi diversi [riuso dei pin]



Sistemi di memoria



□ RAM: Caratteristiche generali

- Esempi di formato di un chip di memoria (altezza x ampiezza)
 - 64K x 8 (64K indirizzi, 8 bit per indirizzo)
 - 32Mx4
 - 128K x 4
 - 2K x 8
- Moduli preassemblati:
 - SIMM (72 pin, Single Inline Memory Module), DIMM (168 pin, Double Inline Memory Module)
 - Più chip di memoria possono essere accoppiati per fornire un bus dati della lunghezza necessaria
 - Es: Modulo DIMM 256 MB: 8 chip per lato ogni chip 32Mx4 $\Rightarrow (8 \times 2) \times 4 \times 32M / 8 = 256 \text{ MB}$



▪ Tempo di ciclo

- Tempo che intercorre fra due operazioni (read o write) su locazioni differenti

Sistemi di memoria



□ ROM: Classificazione

- ROM (programmate in fabbrica)
- PROM (programmabili una volta dall'utente)
- EPROM (cancellabili mediante esposizione a raggi ultravioletti e riprogrammabili)
- EEPROM (cancellabili elettricamente e riscrivibili)
- Tecnologia FLASH (cancellazione per blocchi)

□ Riepilogo tecnologie di memoria

Type	Category	Erasure	Byte alterable	Volatile	Typical use
SRAM	Read/write	Electrical	Yes	Yes	Level 2 cache
DRAM	Read/write	Electrical	Yes	Yes	Main memory
ROM	Read-only	Not possible	No	No	Large volume appliances
PROM	Read-only	Not possible	No	No	Small volume equipment
EPROM	Read-mostly	UV light	No	No	Device prototyping
EEPROM	Read-mostly	Electrical	Yes	No	Device prototyping
Flash	Read/write	Electrical	No	No	Film for digital camera