1. Spiegare la differenza tra interruzioni multiple e interruzioni annidate, discutendo le differenti modalità di trattamento da loro richieste.

Gli interrupt consentono ad altri componenti di interrompere la normale elaborazione della CPU.
Nella gestione di interruzioni multiple, il processore che sta eseguendo l'istruzione di un interrupt e che ne riceve un'altra, ignora quest'ultima fino al termine della prima.
In quel momento l'interruzione pendente verrà presa in carico e potrà essere gestita.
Questa gestione degli interrupt è semplice perché è sequenziale, gli interrupt vengono gestiti in ordine di arrivo, senza tener quindi conto di priorità o tempistiche.
Nella gestione di interruzioni annidate, vengono gestite le priorità.
Il processore che sta eseguendo un'operazione di interrupt riceve un nuovo segnale di interrupt: se quello ricevuto ha priorità maggiore di quello in esecuzione, quello con priorità minore viene sospeso, lo stato viene salvato nella pila di sistema, e viene gestita la nuova richiesta.
Al termine, viene ripristinata la precedente istruzione di interrupt dal punto in cui era stata interrotta e viene portata a termine.

2. Descrivere caratteristiche e le operazioni principali delle memorie a semiconduttore, considerando sia memorie RAM che ROM.

L'elemento base delle memorie a semiconduttore è la cella di memoria che, attraverso due stati stabili, rappresentano i bit 0 o 1.
Nelle celle è possibile scrivere almeno una volta per impostare lo stato, e leggerlo.
Tutte le memorie a semiconduttore sono ad accesso casuale, cioè si può accedere direttamente ad una cella tramite circuiti dedicati.
La RAM (Random-Access Memory) è una memoria volatile, il che significa che per il mantenimento dei suoi dati necessita di alimentazione continua.
Per questo motivo viene utilizzata per memorizzare dati temporaneamente.
È una memoria di lettura e scrittura, i dati vengono cancellati e scritti elettricamente a livello di byte.
La RAM si divide in: DRAM (Dynamic RAM) e SRAM (Static RAM).
La DRAM memorizza i dati come cariche in condensatori.
La presenza o meno della carica nel condensatore rappresenta lo stato 1 o 0.
Queste cariche tendono a disperdersi naturalmente in tempi prestabiliti, anche in presenza di alimentazione.
C'è quindi bisogno di un refresh periodico per mantenere la carica. Per questo vengono dette dinamiche.
Le celle della DRAM presentano una struttura semplice (un solo condensatore per bit) e piccola.
Ha quindi una maggiore densità rispetto alla SRAM, e un minor costo.
Le DRAM sono più lente e vengono generalmente usate nelle memorie principali.
La SRAM utilizza gli stessi elementi del processore (bit memorizzati tramite porte logiche) e non necessita di refresh periodici (gli stati sono stabili).
La struttura delle celle è più complessa (un totale di 6 transistor per cella), e occupano maggior spazio.
Troviamo quindi minor densità rispetto alle DRAM, un costo più elevato, ma una maggiore velocità.
La SRAM viene generalmente utilizzata nella Cache.
Le ROM (Read-Only Memory) sono invece memorie non volatili, cioè mantengono i dati anche in assenza di alimentazione.
I dati sono memorizzati in modo permanente durante la fabbricazione e non possono essere modificati (memorie di sola lettura).
Il processo di fabbricazione è costoso per poche unità, e richiede grande precisione (errata scrittura di un bit compromette l'intero lotto di produzione).
Le ROM vengono utilizzate ad esempio per: subroutine di libreria, programmi BIOS, microprogrammi, ecc.
Le PROM (Programmable Rom) sono memorie ROM che sono programmabili una sola volta, dal fornitore o anche dall'acquirente se in possesso della strumentazione necessaria. (meno costosa rispetto alle ROM per minori volumi di produzione).
Esistono poi memorie specifiche per i casi in cui ci sono principalmente letture, rispetto a scritture, ed è richiesta una memoria non volatile.
-Le EPROM (Erasable PROM) richiedono la cancellazione dei dati prima della scrittura, cancellazione che avviene in maniera totale tramite una luce ultravioletta, attraverso una porta. Processo lento; più costosa della PROM ma può essere aggiornata.
-Le EEPROM (Elettrically Erasable PROM) prevedono una cancellazione a livello di byte, la scrittura risulta molto più lenta della lettura. Più costose delle EPROM ma possono essere aggiornate.
-Memorie Flash: così chiamata per la velocità con la quale possono essere riprogrammate. Come le EEPROM la cancellazione avviene in modo elettrico, ma a livello di blocco e non di byte.

3. Nel contesto di una gerarchia di memoria, spiegare i possibili modi di realizzazione del mapping dei blocchi, discutendo vantaggi e svantaggi.

Ci sono 3 tecniche per l'indirizzamento dei blocchi di memoria:
-associazione diretta (direct mapping): ad ogni blocco di memoria di livello inferiore, viene associata una sola linea di cache.
L'indirizzo viene visto diviso in 3 campi: tag, linea, parola.
I w bit meno significativi (parola) identificano la parola all'interno del blocco.
I restanti s bit identificano uno dei 2^s blocchi in memoria centrale.
Questi bit vengono visti come s-r bit che rappresentano il campo tag, e gli r bit che rappresentano una delle 2^r linee di cache.
Questo tipo di indirizzamento è di semplice realizzazione, la traduzione tra indirizzo ILI (indirizzo di livello inferiore) e indirizzo ILS (indirizzo di livello superiore) è rapido, così come la determinazione di hit o miss (viene individuata la linea nella traduzione dell'indirizzo, e viene avviene poi un confronto tra i due tag).
Lo svantaggio è che due blocchi frequentemente richiesti assegnati ad una stessa linea, genereranno continui swap.

-associazione completa (fully associative): ad ogni blocco della memoria di livello inferiore può essere associata una qualunque linea di cache.
L'indirizzo viene dunque visto diviso in soli due campi: tag e parola.
Abbiamo quindi la massima efficienza di allocazione.
Per stabilire però se un blocco risiede in memoria, tutti i campi tag della cache dovranno essere confrontati in parallelo.
Questo richiede circuiti più complessi per la realizzazione.
Stabilire hit/miss richiede inoltre più tempo
-associazione a gruppi (n-way set associative): le k linee della cache sono raggruppate in v insiemi. In pratica è come avere k associazioni dirette, o v associazioni totali, che operano in parallelo.
Ogni blocco della memoria di livello inferiore può essere associato a qualunque linea dell'insieme di appartenenza.
L'indirizzo viene visto diviso in tre campi: tag, set, parola.
Questo tipo di associazione raggruppa i vantaggi delle due precedenti, e ne diminuisce gli svantaggi.
Abbiamo quindi la miglior associazione complessiva, a fronte di una sopportabile complessità dei circuiti.

4. Descrivere la gestione dell'I/O tramite DMA.

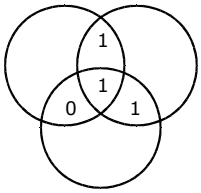
Il DMA (direct memory access) è un modulo hardware addizionale che sostituisce la CPU per la maggior parte delle attività di I/O.
Esso si occupa del trasferimento dei dati da o verso la memoria senza passare attraverso il processore e invia a quest'ultimo un segnale di interrupt quando il trasferimento è completato.
La CPU può essere bloccata prima di richiedere l'uso del bus durante un ciclo istruzione.
Questo non rappresenta un interruzione in quanto la CPU non salva il contesto, si limita a fermarsi per un ciclo di bus.
In questo modo la CPU è coinvolta solo all'inizio e alla fine del trasferimento e nel frattempo può eseguire altre attività.
Prima di delegare una determinata operazione di I/O al DMA, il processore gli comunica le seguenti informazioni:
-se lettura o scrittura
-indirizzo dispositivo interessato
-indirizzo iniziale in memoria del blocco dati coinvolto nell'operazione
-quantità di dati da trasferire.
(Il DMA è connesso al bus di sistema e può accedere al canale dati in 2 modi. Una parola alla volta, sottraendo di tanto in tanto alla CPU il controllo del canale (cycle stealing) o per blocchi, prendendo in possesso il canale per una serie di trasferimenti (burst mode).)
Una configurazione ideale per consumare meno cicli di bus è integrare le funzioni di DMA e I/O.
In questo modo il DMA utilizza il bus di sistema solo per scambiare dati con la memoria, mentre comunica in altri modi con i moduli I/O.
Questo permette anche a DMA di controllare direttamente più dispositivi.
Un configurazione analoga in termini di efficienza è l'utilizzo di un bus I/O separato, che risulta essere una configurazione facilmente espandibile.

5. Spiegare a cosa serve il bus di sistema, com'è strutturato e in che modo viene utilizzato in un calcolatore.

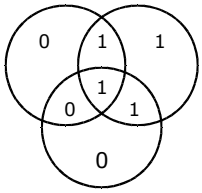
Il bus di sistema è un mezzo di comunicazione condiviso che connette CPU, memoria principale, e moduli di I/O.
È composto da varie linee che trasmettono dati in parallelo (in genere da 50 a qualche centinaio di linee).
Ogni linea trasmette un 1 o uno 0.
Solo un dispositivo alla volta può trasmettere, e l'informazione tra smessa più essere letta da tutti i dispositivi collegati.
Queste linee vengono divise in base alla loro funzione in:
-Linee dati, che permettono ai tre componenti di scambiare dati e istruzioni tra loro (in genere numero linee dati varia tra circa 30 e un centinaio; l'ampiezza di questa linea è importante per l'efficienza del sistema, linee dati piccole causano un accesso ripetuto alla memoria per il prelievo di un dato);
-Linee indirizzi, trasmettono la posizione di un dato in memoria (determina la massima quantità di memoria indirizzabile);
-Linee di controllo, controllano l'accesso e l'uso delle linee dati e indirizzi (i segnali di controllo trasmettono sia comandi, sia informazioni di temporizzazione tra i moduli).
Il bus di sistema opera come segue:
-Se un modulo vuole inviare dati ad un altro, deve ottenere l'uso del bus e successivamente trasferire i dati.
-Se un modulo vuole ricevere dati, deve ottenere l'uso del bus, inviare la richiesta ad un altro modulo tramite le linee indirizzo e controllo, e attendere che i dati vengano inviati.

6. Come funziona il codice di correzione Hamming? Dare un esempio concreto di codifica nel caso di memorizzazione di un insieme di 4 bit.

Il codice di correzione di Hamming è un codice di tipo SEC (Single-Error-Correcting), utile cioè ad identificare e correggere un singolo errore.
Dati M bit da scrivere in memoria, verranno calcolati K bit di controllo (bit di parità) da memorizzare assieme agli M bit.
Il numero di bit di controllo necessari è dato dalla formula $2^K-1>=M+K$.
Per la lettura, tramite gli M bit dati verrà generati altri K bit di controllo, che saranno confrontati bit a bit con i K bit prelevati (XOR esclusivo).
Il risultato del confronto è detto parola sindrome, e identifica la posizione dell'errore.
La parola sindrome può essere così composta:
-tutti 0, indica che non sono presenti errori
-un solo 1, identifica l'errore in un bit di controllo
-più 1, il cui valore numerico identifica la posizione del bit dati errato (correzione con bit complementare)
La parola sindrome e gli M bit dati verranno inviati ad un correttore che emetterà la giusta sequenza degli M bit.
Esempio parola 4 bit:
Il codice di correzione in questo caso può essere graficamente rappresentato da diagrammi di Venn.
Dati i bit 1101, possono essere rappresentati nel seguente modo:



I reparti vuoti verranno riempiti con bit di parità, calcolati in modo che il numero di 1 in ogni cerchio sia pari, quindi:



In presenza di un errore, il bit di parità viene immediatamente rilevato.

7. Nel contesto di una gerarchia di memoria, discutere la modalità e la granularità del trasferimento delle informazioni fra i vari livelli della gerarchia.
Le memorie possono essere organizzate gerarchicamente.
In questo modo, ai livelli più alti troviamo memorie più veloci, meno capienti, e più costose.
La CPU utilizza direttamente il livello più alto.
Più scendiamo di livello, più le memorie sono capienti e meno costose, ma più lente.
Ogni livello della memoria deve contenere i dati del livello inferiore, più altri dati.
I dati tra i livelli vengono trasferiti in blocchi indivisibili.
L'indirizzo di una parola diventa quindi l'indirizzo del blocco, più la posizione della parola all'interno del blocco.

8. Discutere il modo in cui le informazioni sono organizzate in un CD-ROM.
CD-ROM (Compact Disk - Read-Only Memory) è disco ottico di policarbonato rivestito di materiale altamente riflettente (solitamente alluminio).
Le informazioni vengono registrate tramite un laser ad alta intensità che crea dei pit ("pozzetti") sullo strato di policarbonato, che vengono poi rilevati da un laser a bassa frequenza.
Il laser rileva la variazione di intensità della luce riflessa quando incontra un pit.
Le aree tra i pit sono dette land.
L'inizio o la fine di un pit rappresentano un 1, le aree piane lo 0.
Le informazioni sono organizzate in un'unica traccia a spirale, che parte dal centro del disco fino al bordo.
Il disco è diviso in settori della stessa dimensione lungo tutta la traccia, saranno quindi uguali sia al centro che alla periferia del disco.
I dati sono organizzati in una sequenza di blocchi, ciascuno contenente i seguenti campi:
-Sync: identifica l'inizio di un blocco, byte di 0 e 1 ordinati
-ID: identifica la posizione del blocco nella traccia, e il modo (modo 0: campo dati vuoto; modo 1: campo dati + campo auxiliary contenente codice correzione d'errore; modo 2: campo data + auxiliary contenente solo dati)
-Data: dati utente
-Auxiliary: codice di correzione d'errore nel modo 1, dati utente aggiuntivi nel modo 2
I pit sono letti dal laser a velocità lineare costante, facendo ruotare il disco a velocità variabile.

9. Descrivere in dettaglio il ciclo di esecuzione con trattamento delle interruzioni.
La prima fase è il calcolo dell'indirizzo dell'istruzione da eseguire.
Durante la successiva fase di fetch, l'istruzione viene prelevata dalla memoria principale, l'indirizzo posto sul registro IR (Instruction Register), e viene incrementato il valore del PC con l'istruzione immediatamente successiva in memoria.
L'istruzione prelevata viene quindi decodificata e analizzata per determinare l'operazione da eseguire e gli operandi (se più di uno) necessari.
Si determinano gli indirizzi degli operandi, che vengono poi prelevati dalla memoria (o periferica).
Si esegue l'operazione indicata dall'istruzione e il risultato viene scritto in memoria (o inviato alla periferica).
A questo punto, la CPU controlla se ci sono degli interrupt pendenti:
-in caso negativo, viene prelevata la prossima istruzione
-in caso affermativo, la CPU sospende la corrente esecuzione e ne salva il contesto (cioè l'indirizzo della prossima istruzione e gli altri dati necessari). Imposta il registro PC all'indirizzo di partenza della routine per la gestione degli interrupt. Procede quindi al fetch e legge la prima istruzione del programma di gestione dell'interrupt. Quando questa routine termina, la CPU riprende l'esecuzione del programma utente dal punto dell'interruzione.

~~10.~~ Spiegare in dettaglio le differenze tra un modulo di memoria DRAM ed un modulo di memoria SRAM, discuterne vantaggi e svantaggi.
La memoria RAM (Random-Access Memory) si divide in Dynamic RAM (DRAM) e Static RAM (SRAM).
Sono memorie di lettura/scrittura volatili, hanno quindi bisogno di alimentazione per mantenere i dati.
Per questo vengono utilizzate per la memorizzazione temporanea dei dati.
Sono memorie ad accesso casuale, cioè l'accesso ai dati avviene in modo diretto tramite circuiti dedicati.
Le DRAM memorizzano i dati mediante cariche in condensatori.
La presenza o meno della carica identifica lo stato 1 o 0.
Le cariche nei condensatori tendono a disperdersi naturalmente in tempi predefiniti, anche in presenza di alimentazione.
È quindi necessario un refresh periodico per il mantenimento del dato corretto.

Questo il motivo del termine Dynamic.
Le celle presentano una struttura semplice, un condensatore per bit, e sono più piccole di quelle della SRAM.
La densità è quindi maggiore, e si ha un minor costo.
Sono però più lente, inoltre necessitano di circuiti per il refresh periodico.
Le SRAM utilizzano invece gli stessi componenti del processore (bit memorizzati tramite porte logiche).
Non hanno quindi bisogno di refresh delle cariche, ma presentano stati stabili.
La struttura della cella è più complessa e occupa più spazio (un totale di 6 transistor per bit).
La densità è dunque minore e il costo più elevato.
Sono però più veloci, e vengono generalmente utilizzate nelle Cache.
La DRAM invece viene di solito usata nelle memorie principali.

11. Discutere le ragioni per cui è stato sviluppato il sistema RAID. Si descrivano inoltre i vari livelli.

RAID è l'acronimo di Redundant Array of Indipendent (o Inexpensive) Disks.
Il sistema RAID è stato sviluppato principalmente per tre ragioni, che non necessariamente devono coesistere:
-le altre componenti erano molto più veloci del disco, quindi si è pensato di aumentare le prestazioni premettendo un accesso parallelo a più dischi
-avere dei dati ridondanti, quindi maggiormente preservati da possibili malfunzionamenti hardware
-avere un maggior spazio di memorizzazione in modo economico, cioè senza dover spendere tempo e denaro per lo sviluppo di dispositivi più capienti
Il sistema RAID permette al sistema operativo di vedere un insieme di dischi fisici come un singolo dispositivo logico ed i dati vengono distribuiti sui vari dispositivi fisici.
Il RAID è suddiviso in 7 livelli (dallo 0 al 6) non gerarchici che definisco diverse scelte architetturali.
-RAID 0
Il RAID di livello 0, a differenza degli altri, non include la ridondanza.
In dati sono distribuiti su tutti i dischi in strisce, che possono essere blocchi, settori, o altro.
Le strisce sono distribuite a rotazione (round robin striping) su tutti i dischi dell'array.
Richieste multiple di dati hanno quindi scarsa probabilità di coinvolgere lo stesso disco.
Le richieste possono quindi essere gestite in parallelo.
Lo svantaggio è che il guasto di un disco causa la perdita dei suoi dati.
-RAID 1
Differisce dal RAID 0 per la ridondanza.
I dati sono distribuiti su tutti i dischi dell'array in strisce, a rotazione (round robin).
La ridondanza è ottenuta duplicando i dati (si parla di mirrored disks).
Lettura/scrittura avvengono quindi in parallelo su entrambi i dischi.
Le letture verranno soddisfatte dal disco con lettura più veloce.
Le scritture sono dettate dal disco più lento.
Il recupero dell'informazione dopo un guasto è semplice, basta sostituire il disco e copiare i dati.
Non ci sono quindi periodi di inattività.
Uno svantaggio è il costo (necessario spazio doppio per la memorizzazione).
Questo sistema fornisce un backup in tempo reale, è quindi utile per software di sistema, e file/dati altamente critici.
-RAID 2 (non commercializzato)
Utilizza tecniche di accesso in parallelo.
I dischi sono sincronizzati: in ogni disco la testina si trova nella stessa posizione.
Striping dei dati.
Strisce piccole (spesso singolo byte o parola).
Viene calcolato un codice di correzione d'errore sui corrispondenti bit di ogni disco dati.
I bit del codice di correzione vengono memorizzati in nelle corrispondenti posizioni di bit, su più dischi di parità.
Solitamente si adotta il codice di Hamming.
RAID 2 richiede meno dischi di RAID 1, ma rimane comunque piuttosto dispendioso.
-RAID 3
Simile al RAID 2 (striping dei dati, accesso in parallelo e dischi sincronizzati, strisce piccole).
Al posto del codice di correzione di errore, viene calcolato un bit di parità tra i bit nella stessa posizione sui vari dischi.
Avremo quindi un solo disco ridondante.
In caso di guasto di un disco, il recupero sarà semplice (tramite dati sui dischi rimanenti e bit su disco di parità)
Lo svantaggio è che i dischi operano in parallelo, quindi non indipendentemente.
-RAID 4 (non commercializzato)
Viene introdotta l'indipendenza nell'operatività dei dischi, richieste di I/O vengono quindi gestite più efficacemente.
Le strisce diventano più grandi.
Viene utilizzata una striscia di parità bit a bit, che viene memorizzata nella corrispondente striscia di un parity disk.
Lo svantaggio è che ogni operazione di scrittura deve coinvolgere anche il disco di parità, che può diventare un collo di bottiglia.
-RAID 5
Simile al RAID 4.
Quindi, striping dei dati, dischi che operano indipendentemente, strisce più grandi, utilizzata striscia di parità bit a bit.
Al posto del parity disk, le strisce di parità sono distribuite su tutti i dischi tramite il round robin (evita il collo di bottiglia).
Il numero di dischi totale sarà quindi N+1.
-RAID 6
Il livello 6 aumenta l'affidabilità calcolando bit di parità in due distinti metodi.
Parità è memorizzata in blocchi separati, su dischi differenti.
È richiesto quindi un totale di N+2 dischi.
C'è quindi una maggiore affidabilità (per la perdita dei dati, occorre guasto su 3 dischi).
La scrittura diventa però più lenta perché devo aggiornare due informazioni di parità ogni volta.

12. Descrivere in dettaglio il ciclo completo di fetch/execute delle istruzioni.

IR (instruction register): registro che contiene il codice operativo dell'istruzione correntemente in esecuzione.
MBR (memory buffer register): contiene una parola che deve essere immagazzinata in memoria o è pervenuta dalla memoria.
MAR (memory address register): contiene l'indirizzo della parola di memoria in cui scrivere il contenuto di MBR o che deve essere trasferita in MBR.
PC (program counter): contiene l'indirizzo della prossima coppia di istruzioni da caricare dalla memoria.
AC (accumulator) e MQ (multiplier quotient): contengono temporaneamente gli operandi e i risultati parziali delle operazioni della ALU.
Durante il ciclo di prelievo (fetch cycle), il codice operativo della successiva istruzione viene caricato nell'IR e la porzione di indirizzo viene caricata nel MAR.
Questa istruzione può essere ottenuta dalla memoria caricando una parola nel MBR, e a seguire nell'IR e nel MAR.
Quando il codice operativo è stato registrato nell'IR, ha inizio il ciclo di esecuzione (execute cycle).
Il circuito di controllo interpreta il codice operativo ed esegue l'istruzione, impostando i segnali di controllo appropriati per trasferire i dati o per fare eseguire un'operazione della ALU.

13. Descrivere in dettaglio le memorie DRAM.

Le DRAM (Dynamic RAM) memorizzano i dati mediante cariche in condensatori.
La presenza o meno della carica identifica lo stato 1 o 0.
Le cariche nei condensatori tendono a disperdersi naturalmente in tempi predefiniti, anche in presenza di alimentazione.
È quindi necessario un refresh periodico per il mantenimento del dato corretto.
Questo il motivo del termine Dynamic.
Le celle presentano una struttura semplice, un condensatore per bit, e sono più piccole di quelle della SRAM.
La densità è quindi maggiore, e si ha un minor costo.
Sono però più lente, inoltre necessitano di circuiti per il refresh periodico.
La DRAM invece viene di solito usata nelle memorie principali.
(La linea d'indirizzo viene attivata quando si deve leggere o scrivere il bit della cella.
Il transistor si comporta da corto circuito (non si oppone al flusso di corrente) quando c'è tensione sulla linea d'indirizzo e da circuito aperto (impedisce il flusso di corrente) quando non c'è tensione sulla linea d'indirizzo.)
Per le operazioni di scrittura scrittura, si applica tensione alla linea di bit (a seconda dell'intensità si determina se bit 1 o 0). Quindi si applica un segnale alla linea d'indirizzo, che trasferisce la carica al condensatore.
Per quelle di lettura, la linea di indirizzo è selezionata, il transistor si accende per far transitare la carica immagazzinata nel condensatore.
La carica percorre una linea di bit collegata a un amplificatore, che confronta la tensione con un valore di riferimento e determina se la cella contiene un 1 o uno 0.
Poiché la lettura dalla cella scarica il condensatore, per completare l'operazione si deve ripristinare la carica.

14. Nel contesto di una gerarchia di memoria, illustrare le possibili tecniche (incluse quelle che coinvolgono il compilatore) che si possono adottare per tentare di minimizzare il numero di miss.

Nel contesto di una gerarchia di memoria, i miss possono essere categorizzati in tre tipi:
-miss di primo accesso: inevitabili e non riducibili
-miss per capacità insufficiente: quando la cache non può contenere tutti i blocchi necessari all'esecuzione del programma
-miss per conflitto: nell'associazione diretta o a gruppi quando più blocchi possono essere mappati in uno stesso gruppo.
Le strategie che si possono adottare per diminuire il numero di miss sono:
-maggiore dimensione del blocco (che però causa aumento miss per conflitto-mano blocchi disponibili)
-maggiore associatività (che però causa incremento del tempo di localizzazione in gruppo; inoltre soggetta alla regola 2:1, in cache a N blocchi, probabilità miss pressoché uguale che in cache a N/2 blocchi con associazione a 2 vie)
-utilizzo cache multilivello, cioè una gerarchia di cache. Vi sarà un cache piccola e molto veloce sullo stesso chip della CPU, detta cache on-chip. Accesso molto veloce. Vi saranno poi altre cache (L2-L3), più grandi e leggermente più lenti, associate. Le cache sono connesse tra di loro in modo indipendente dal bus di sistema, così da non pesare su di esso. Nonostante l'aumento dei dispositivi tra CPU e memoria centrale, vi è un aumento prestazionale rispetto alla connessione diretta della L1 alle memoria centrale, in quanto, grazie alla localizzazione spaziale, vi sarà un alto numero di hit dentro i dati nella L3, la quale, in caso di miss potrà inviare i dati sino alla L1 in un tempo estremamente ridotto rispetto a quello che farebbe la memoria centrale.
-separazione tra cache dati e cache istruzioni, così da rendere indipendente la scrittura/lettura dei dati dalla singola lettura delle istruzioni
-ottimizzazione degli accessi mediante compilatori. Tale compilatore deve essere in grado di: utilizzare in modo ottimale l'architettura della cache, ad esempio se multilivello; posizionamento accurato delle procedure ripetitive; incrementare la località spaziale mediante la fusione di vettori in strutture ottimali; incrementare la località spaziale mediante l'ottimizzazione di iterazioni annidate.

15. Descrivere l'organizzazione e formattazione dei dati nei dischi rigidi.

I dati sul disco sono memorizzati in anelli concentrici, chiamate tracce, della stessa larghezza della testina.
Un disco dispone di migliaia di tracce.
Tracce adiacenti sono separate da spazi (gaps), per minimizzare errori dovuti di disallineamento della testina o interferenza tra i campi magnetici.
Ogni traccia contiene lo stesso numero di bit.
I dati nelle tracce sono organizzati in settori, che sono l'unità minima di trasferimento.
La dimensione minima di un blocco coincide con un settore (ogni blocco può avere quindi più settori).
Ci sono un centinaio di settori per traccia, di lunghezza fissa o variabile (generalmente di lunghezza fissa di 512 byte).
Settori adiacenti sono separati da spazi.

L'accesso è diretto: la testina si sposta direttamente su una traccia, e attende il passaggio del settore richiesto.
Un problema è che i bit più vicini al centro ruotano a una velocità relativa più bassa dei bit più esterni.
Per poter leggere tutti i bit alla stessa velocità, si può aumentare lo spazio tra i bit nelle tracce più esterne.
Le informazioni sono quindi lette alla stessa velocità facendo ruotare il disco a velocità angolare costante.
Si ha però densità minore man mano che ci si allontana dal centro, e quindi spreco di spazio.
Per incrementare la densità, si può utilizzare la tecnica di registrazione a più zone.
La superficie viene ripartita in zone, ciascuna con lo stesso numero di bit.
Le aree più lontane dal centro saranno quindi composte da più zone.
La densità è maggiore al costo di circuiti leggermente più complessi.
Tramite la formattazione siamo in grado di localizzare l'inizio di una traccia e la posizione di un settore al suo interno.
Il disco viene quindi inizializzato con informazioni disponibili solo al suo sistema di controllo e non all'utente.

16. Descrivere in dettaglio la gestione da programma I/O.

Nell'I/O da programma, è la CPU che si fa carico di tutta la gestione.
Il funzionamento è il seguente:
-CPU invia al modulo di I/O un comando (invia anche indirizzo della periferica + identificatore se più di una)
-Modulo esegue l'azione richiesta e imposta i bit appropriati nel suo registro di stato
-Modulo si interfaccia a periferica fino al completamento dell'istruzione ricevuta
-il Modulo non avvisa la CPU del completamento della richiesta, è la CPU che controlla lo stato del Modulo periodicamente (alla fine del ciclo di un'istruzione)
Ci sono 4 tipi di comandi che il processore può inviare al modulo I/O:
-controllo, avvia una periferica e le dice cosa fare
-test, testa le condizioni di stato dei moduli di I/O
-lettura, ottiene dati dalla periferica attraverso il modulo I/O
-scrittura, impone al modulo di trasmettere tramite bus i dati alla periferica
Si ha quindi uno spreco di tempo per la CPU che deve attendere il completamento dell'istruzione e interrogare continuamente lo stato del modulo.

17. Descrivere in cosa consiste l'architettura Von Neumann.

Quasi tutti gli odierni calcolatori si basano sul modello di John Von Neumann.
L'architettura detta di Von Neumann si basa su tre caratteristiche principali:
-Dati e istruzioni risiedono in un'unica memoria di lettura e scrittura.
-Ogni dato/istruzione è accessibile per indirizzo.
-L'esecuzione delle istruzioni avviene in modo sequenziale, da un indirizzo di memoria a quello immediatamente successivo.
Per eseguire un programma dobbiamo costruire i componenti logici in modo da ottenere il risultato desiderato.
Questo tipo di architettura è detta programma cablato (o hardware), in quanto non può essere modificato.
È un sistema non flessibile, può eseguire solo operazioni determinate.
Tramite circuiti generici accetta segnali di controllo e produce risultati.
Per ogni nuovo programma, basta dare i giusti segnali di controllo.
(La programmazione software è molto più facile perché, invece che ridefinire ogni volta l'hardware, basta fornire una nuova sequenza di codici (ossia una nuova istruzione).)

18. Nel contesto di una gerarchia di memoria spiegare perché la memoria viene suddivisa in blocchi e, relativamente alle prestazioni della cache, discutere pregi e difetti dell'adozione di una dimensione di blocco elevata.

La cache contiene una copia di parti della memoria principale.
Quando la CPU legge una parola, controlla prima la cache.
In caso affermativo la parola viene consegnata al processore.
In caso negativo, la parola trasferita nella cache e poi letta dal processore. Quindi il processo subisce un rallentamento.
Per il principio di località dei riferimenti, è probabile che le future richieste della CPU riguarderanno parole nelle vicinanze.
Per questo motivo, i dati tra i livelli vengono trasferiti in blocchi indivisibili.
L'indirizzo di una parola diventa quindi l'indirizzo del blocco, più la posizione della parola all'interno del blocco.
Nel caso di un blocco con dimensione elevata:
-si guadagna in termini di costi ed è un buon metodo per la località spaziale
-si perde in termini di velocità di trasferimento e di prestazioni (maggiori miss per conflitto a causa dello spazio ridotto).
-Blocchi più larghi riducono il numero di blocchi in cache (piccolo numero di blocchi porta alla sovrascrittura dei dati in fasi brevi)
-quando i blocchi sono troppo grandi, una parola è lontana dalla parola attuale, diminuisce quindi la probabilità che venga richiesta a breve.

19. Spiegare cosa sono gli errori soft, come ovviare a tali errori e fare eventualmente un esempio.

Le memorie a semiconduttore sono soggetti ad errori che possono essere guasti hardware, permanenti e irreversibili, oppure errori software, casuali e non permanenti, alterano il contenuto di una cella senza danneggiarla.
Errori software possono essere generati per esempio da problemi di alimentazione o da particelle alfa.
Queste particelle derivano dal decadimento radioattivo, problema comune a praticamente tutti i materiali.
Questi errori software necessitano ovviamente di una risoluzione, che viene data dai codici di correzione d'errore.
Il funzionamento di questi codici è il seguente:
Quando una parola di M bit deve essere memorizzata, tramite una funzione f viene generato un codice di K bit, che verrà memorizzato assieme alla parola.
Quindi la parola memorizzata sarà di M+K bit.
Quando una parola deve essere letta, gli M bit della parola vengono utilizzati per generare dei nuovi K bit, che verranno confrontati bit a bit con i K bit prelevati.

Il controllo genera i seguenti risultati:
-non vengono rilevati errori, si procede quindi con il prelievo della parola.
-viene rilevato un errore e lo si può correggere. I bit dati e i bit di correzione dell'errore vengono inviati ad un correttore che emette il corretto insieme di bit.
-viene rilevato un errore ma non è possibile correggerlo. Viene inviato un segnale di errore.
Un semplice codice a correzione d'errore è il codice di Hamming.
È un tipo di codice SEC (Single Error Correcting), in grado cioè di rilevare e correggere un solo errore.
Il funzionamento del codice di Hamming è quello sopra descritto.
In base al numero di bit dati, vengono generati dei bit di controllo.
Quanti bit di controllo è determinato attraverso la formula $2^K-1>=M+K$, quindi ad esempio a una parola da 4 bit saranno associati 3 bit di controllo, a una parola da 8 bit saranno associati 4 bit di controllo, e così via.
Il valore di ciascuno dei bit di controllo, sarà il bit di parità dato risultato dello XOR di ogni bit dato che presenta un 1 nella stessa posizione del numero di posizione del bit di controllo.
I K bit di controllo letti dalla memoria verranno confrontati con i K bit generati dopo il prelievo (XOR esclusivo).
Il loro confronto genera la parola sindrome, che darà le seguenti informazioni:
-se composta da tutti 0, significa che non sono presenti errori.
-se composta da un 1, identifica l'errore in uno dei bit di controllo
-se composta da più 1, il loro valore numerico indicherà la posizione del bit errato (correzione con bit complementare).
Per migliorare l'efficienza alcuni sistemi utilizzano in aggiunta anche un codice SEC-DED (Double Error Detecting).
Questo codice è utile nel caso di 2 errori (evento poco probabile).
Il codice SEC-DED richiede un bit aggiuntivo e viene calcolato tenendo conto della parità globale.

20. Descrivere le politiche di scrittura write through e write back evidenziando differenze, vantaggi e svantaggi.
Quando un dato in cache viene aggiornato, bisogna aggiornare anche i livelli inferiori di memoria.
Con la politica write-through la scrittura viene eseguita in modo simultaneo.
Questo permette di avere i dati coerenti tra i vari livelli di memoria, ma c'è un continuo traffico per scritture frequenti su uno stesso blocco, con un conseguente possibile collo di bottiglia.
Con la politica write-back la scrittura in memoria principale è differita e avviene solo nel momento in cui il blocco deve essere sostituito in cache. Occorre anche identificare (attraverso un dirty bit) i dati modificati.
Questo consente di ottimizzare il traffico del bus, ma causa periodi di incoerenza tra le memorie (causa di problemi in sistemi multiprocessore con memoria condivisa).
(Possibili soluzioni possono essere:
-monitoraggio del bus con write-through: controllori cache monitorano le linee indirizzi e intercettano modifiche locazioni condivise
-hardware aggiuntivo, che rileva modifiche alla memoria principale e modifica tutte cache di conseguenza
-solo una porzione di memoria può essere condivisa e noncacheable. Gli accessi alla memoria condivisa genereranno sempre dei miss, perché non viene mai copiata nella cache. Essa può essere identificata via hardware o tramite indirizzi riservati.)

21. Descrivere in che modo vengono gestite le interruzioni (sia per la componente hardware che per quella software) nel caso di I/O input driven.
Il vantaggio principale della gestione con interrupt è l'abolizione dell'attesa da parte della CPU, che non dovrà continuamente interrogare lo stato del modulo.
Sarà infatti compito del Modulo di I/O avvisare la CPU a completamento dell'operazione richiesta.
A livello hardware, quando una periferica completa un'azione di I/O avviene la seguente sequenza di eventi:
-il dispositivo invia un interrupt al processore
-il processore controlla l'eventuale presenza di interrupt al termine di un ciclo istruzione
-se c'è riscontro, invia un segnale di riconoscimento al dispositivo che ha inviato tale interrupt che rimuoverà quindi il proprio segnale
-il processore salva il contesto (indirizzo prossima istruzione, dati registri, e altri dati utili) in cima alla pila di sistema
-il processore scrive nel PC l'indirizzo della prima istruzione della routine di gestione dell'interrupt
-dopodiché si esegue il fetch dell'istruzione e il controllo viene quindi trasferito al programma di gestione dell'interrupt.
A livello software, i successivi passaggi sono i seguenti:
-la routine dell'interrupt salva le restanti informazioni di stato del processo
-elabora l'interrupt
-quando l'elaborazione è completa il contesto precedente all'interrupt viene recuperato dalla pila di sistema e i valori ripristinati

22. Descrivere la differenza tra architettura e organizzazione di un calcolatore.
L'architettura di un calcolatore è l'insieme degli attributi visibili al programmatore, che hanno cioè un impatto diretto sull'esecuzione logica di un programma (esempi sono il set di istruzioni, l'indirizzamento della memoria, bit usati per rappresentare un dati).
L'organizzazione fa riferimento alle unità operative e alle loro connessioni che realizzano specifiche architetturali, rappresenta quegli elementi trasparenti al programmatore (esempi sono i segnali di controllo, interfacce tra cpu e periferiche, tecnologia delle memorie).

23. Descrivere la struttura e il funzionamento di un modulo di memoria SRAM.
Le SRAM (Static RAM) utilizzano gli stessi componenti del processore (bit memorizzati tramite porte logiche).
A differenza delle DRAM, non hanno quindi bisogno di refresh delle cariche, ma presentano stati stabili.
La struttura della cella SRAM è più complessa e occupa più spazio (un totale di 6 transistor per bit).
La densità è dunque minore e il costo più elevato.
Sono però più veloci, e vengono generalmente utilizzate nelle Cache.
Quattro transistor (T_1, T_2, T_3, T_4) sono connessi in modo da costituire uno stato logico stabile.

Nello stato logico 1, il punto C_1 è alto e il punto C_2 è basso; in questo stato, T_1 e T_4 sono spenti, mentre T_2 e T_3 sono accesi.
Nello stato logico 0, il punto C_1 è basso e il punto C_2 è alto; in questo stato, T_1 e T_4 sono accesi, mentre T_2 e T_3 sono spenti.
Entrambi gli stati sono stabili finché la cella è alimentata (con corrente continua).
A differenza delle DRAM, per mantenere i dati non è richiesto il refresh.
Come nella DRAM, la linea di indirizzo viene usata per aprire o chiudere un interruttore.
La linea di indirizzo controlla due transistor (T_5 e T_6).
Quando a questa linea viene applicato un segnale, i due transistor vengono accesi, consentendo la lettura o la scrittura. Per un'operazione di scrittura, si applica il valore desiderato del bit alla linea B, e il valore negato alla linea .
Questo forza i quattro transistor (T_1, T_2, T_3, T_4) allo stato corretto.
In lettura, il valore viene letto dalla linea B.

~~24.~~ Nel contesto di una gerarchia di memoria, spiegare come funziona la politica di scrittura write back. Discutere criticamente i problemi che possono insorgere nell'adottarla.
La politica di scrittura write-back è una tecnica alternativa alla write through che evita il collo di bottiglia creato da quest'ultima minimizzando le scritture in memoria e applicando gli aggiornamenti solo nella cache.
Occorre inoltre identificare (attraverso un dirty bit) i dati modificati.
Il problema di questa tecnica è che parti delle memorie di livello inferiore non sono aggiornate.
Questa tecnica causa problemi in sistemi multiprocessore con memoria condivisa.
Una modifica dei dati in una cache può invalidare tutti i dati nella memoria centrale e anche quelli nelle altre cache eventualmente connesse al bus.
(Possibili soluzioni possono essere:
-monitoraggio del bus con write-through: controllori cache monitorano le linee indirizzi e intercettano modifiche locazioni condivise
-hardware aggiuntivo, che rileva modifiche alla memoria principale e modifica tutte cache di conseguenza
-solo una porzione di memoria può essere condivisa e noncacheable. Gli accessi alla memoria condivisa genereranno sempre dei miss, perché non viene mai copiata nella cache. Essa può essere identificata via hardware o tramite indirizzi riservati.)

a. Descrivere differenze tra bus singoli e bus multipli.
In generale, più dispositivi sono collegati ad un bus.
Più il bus è lungo, più le prestazioni ne soffriranno in quanto si verificheranno maggiori ritardi e possibili congestioni.
L'uso di bus multipli, generalmente disposti gerarchicamente, è una soluzione a questo problema.
Una struttura tradizionale consiste in un bus di memoria, che collega CPU e cache.
La cache è dunque collegata sia al bus locale che al bus di sistema.
La memoria principale è collegata al bus di sistema, questo permette la sua comunicazione con le periferiche senza il coinvolgimento del processore.
Un bus di espansione può essere utile come livello gerarchico più basso, per supportare più periferiche isolando il traffico loro e il modulo di I/O dal bus di sistema.

b. Descrivere differenze tra linee dedicate e linee multiplexate.
Le linee dedicate sono destinate permanentemente ad un funzione/componente.
Un esempio, sono le linee separate per dati e indirizzi.
Vantaggi: minore contesa bus.
Svantaggi: dimensioni e costo.
Le linee multiplexate trasmettono per esempio sia dati che indirizzi.
La trasmissione avviene utilizzando la linea di controllo Address Valid.
Ogni modulo legge l'indirizzo e determina se è lui il destinatario.
Una volta stabilito, l'indirizzo viene rimosso dal bus e i dati trasmessi.
Vantaggio multiplexing: numero inferiore di linee, quindi risparmio spazio e abbattimento costi.
Svantaggio: all'interno di ogni modulo richiesti circuiti più complessi. Prestazioni ridotte.

~~c.~~ Nel caso di trasmissione sincrona dei dati, spiegare come sono coordinati gli eventi (temporizzazione) in un bus.
La temporizzazione di riferisce al modo in cui gli eventi sono coordinati sul bus.
Può essere sincrona o asincrona.
Sincrona: una linea di clock sincronizza e determina gli eventi.
Bus ha una linea di clock che trasmette sequenza alternata di 0 e 1 di uguale durata.
La trasmissione della coppia 0-1 viene detta ciclo di clock e definisce un intervallo di tempo.
Tutti i dispositivi connessi al bus possono leggere la linea di clock.
Tutti gli eventi partono all'inizio di un ciclo di clock.

Vantaggio: più semplice. Svantaggio: meno flessibile; il sistema non può avvantaggiarsi di migliorie nelle prestazioni dei dispositivi.
Con la temporizzazione asincrona, l'occorrenza di un evento dipende dall'occorrenza di un evento precedente.

d. Spiegare cosa si intende per località dei riferimenti.
Località dei riferimenti è un principio che consente di ottimizzare il tempo di accesso ai dati.
Per il principio di linearità dei riferimenti, i riferimenti alla memoria da parte del processore tendono a raggrupparsi, quindi accessi a indirizzi contigui sono più probabili (località spaziale).
Sempre per lo stesso principio, la zona di accesso più recente è quella con maggiore probabilità di permanenza (località temporale).

e. Illustrare le politiche di rimpiazzo dei blocchi.
Per poter scrivere un nuovo blocco nella cache, bisogna sostituire uno dei suoi blocchi.
Con l'indirizzamento diretto, non sorgono problemi in quanto ogni blocco può essere scritto in una sola linea.
Per le altre due tecniche, gli algoritmi di rimpiazzo più comuni sono:
-Casuale, la scelta del blocco da sostituire è casuale.
-FIFO: First-In-First-Out, viene sostituito il blocco che risiede in memoria da più tempo. Non è una buona tecnica nel caso in cui il blocco che permane da più tempo sia anche molto utilizzato.
-LFU: Least Frequently Used, viene sostituito il blocco con meno accessi. Non è una buona tecnica nel caso di un blocco abbastanza recente, che potrebbe prossimamente essere usato molto.
-LRU: Least Recently Use, viene sostituito il blocco usato meno recentemente (preserva località temporale). Tecnica complessivamente più efficiente.

f. Descrivere il meccanismo di lettura e scrittura in un disco magnetico.
Un disco è un piatto circolare di vetro (chiamato substrato) rivestito di materiale magnetizzabile (ossido di ferro).
I dati vengono memorizzati e recuperati tramite una testina (bobina conduttiva).
Molti sistemi utilizzano una testina separata per la lettura e la scrittura.
Durante lettura/scrittura, la testina è ferma mentre il disco ruota.
Per la scrittura, la corrente che fluisce nella bobina produce un campo magnetico.
Vengono inviati impulsi alla testina, che memorizza 0 o 1 sul disco sotto forma di campi magnetici (con direzione opposta).
La testina di scrittura è fatta di materiale altamente magnetizzabile, è a forma di ferro di cavallo stilizzato attorno a cui è avvolto qualche giro di filo conduttore che fa transitare la corrente.
La direzione della corrente indica l'orientamento del campo magnetico.
La testina di lettura separata è posta vicino a quella di lettura.
Essa comprende un sensore magnetoresistivo parzialmente schermato (la schermatura permette di ridurre l'influenza di campi magnetici adiacenti).
La resistenza elettrica dipende dalla direzione del campo magnetico sottostante (aumenta o diminuisce la resistenza in funzione del campo magnetico).

g. Descrivere le caratteristiche fisiche di un disco magnetico.
1. La testina può essere fissa o mobile
Nel primo caso, siamo in presenza di un braccio fisso con testina/e di lettura/scrittura per ogni traccia.
Nel secondo caso, la testina di lettura/scrittura è una per faccia del disco, ed è montata su un braccio mobile.
2. Disco fisso o removibile.
Nel primo caso, il disco è montato permanentemente in un telaio, nel secondo caso può essere rimosso e sostituito con un altro disco (la capacità di memorizzazione è quindi potenzialmente illimitata).
3. Facce/piatti multipli.
Nel primo caso, la parte magnetizzabile è applicata ad entrambe le facce del disco.
Nel secondo caso, vengono utilizzati più piatti allineati verticalmente a distanza di qualche centimetro.
Le testine sono montate in un unico braccio, e in ogni momento la testina di ogni faccia è allineata per essere alla stessa distanza dal centro del disco.
L'insieme delle tracce dei vari piatti nella stessa posizione vengono detti cilindri.
I dati sono quindi distribuiti su cilindri: questo permette di ridurre lo spostamento delle testine, e di eseguire lettura/scrittura in parallelo.

h. Descrivere le caratteristiche di un disco Winchester.
Il disco Winchester venne sviluppato da IBM ed è oggi un sistema universale ed economico.
I dischi vengono assemblati assieme alle testine (ermeticamente sigillati per non permettere il passaggio di agenti contaminanti).
Le testine, dette foil (foglia), sono posizionate più vicine al disco.
A disco fermo, esse sono leggermente appoggiate alla superficie.
Quando il disco ruota, la pressione d'aria generata fa allontanare la foglia dalla superficie.
Data la vicinanza della testina al disco, avremo una minore interferenza minore tra tracce/settori, e la possibilità quindi di aumentare la densità di memorizzazione.

i. Descrivere i parametri prestazionali di un disco magnetico.
Il disco ruota ad una velocità angolare costante.
Per poter leggere un dato, la testina deve trovarsi nella giusta traccia e attendere il passaggio del settore desiderato.
In presenza di una testina mobile, la selezione della traccia implica il movimento della testina.
Il tempo richiesto per questa operazione viene detto tempo di posizionamento (seek time).
Il tempo di posizionamento è difficilmente riducibile (in genere tra 5-20 ms) e dipende dalla dimensione del disco.
Quando la testina è posizionata, l'attesa del settore desiderato viene detta latenza rotazionale (latency).
La latenza dipende dalla velocità di rotazione del disco.

Tempo di posizionamento+latenza rotazionale equivalgono al tempo di accesso.
Una volta che la testina è in giusta posizione, l'operazione di lettura/scrittura viene eseguita.
Questo implica un tempo di trasferimento dei dati, dato dalla formula:

$$T = \frac{b}{r*N}$$

dove: b = byte da trasferire
N = byte per traccia
r = velocità rotazionale per sec.

j. Descrivere un nastro magnetico magnetico.

Un nastro magnetico è un nastro flessibile composto da poliestere e rivestito di materiale magnetizzabile.
I dati sono registrati lungo tracce longitudinali che vengono lette in parallelo.
Lettura e scrittura avvengono in modo seriale/sequenziale, cioè un bit dopo l'altro, per tutta la lunghezza del nastro.
La registrazione viene detta a serpentina, arrivati alla fine del nastro questo non viene riavvolto ma si passa alla traccia successiva invertendo la direzione di registrazione.
I nastri sono dispositivi molto economici e capienti, ma molto lenti.
Trovano largo impiego per il backup dei dati e archiviazione.

k. Descrivere le funzioni di un modulo I/O.

I/O è il terzo elemento chiave del sistema.
Permette al calcolatore di comunicare con una grande varietà di periferiche.
I moduli di I/O si interfacciano con CPU e memoria tramite il bus di sistema, con le periferiche tramite linee su misura.
Dispositivi esterni si classificano in dispositivi:
-comprensibili all'uomo, come video o stampanti
-comprensibili alla macchina, come dischi magnetici
-di comunicazione, come modem
Un modulo di I/O è composto da:
-una logica di controllo che riceve segnali di controllo, e invia segnali di stato su richiesta della CPU.
-buffer, che invia/riceve dati e li memorizza temporaneamente
-trasduttore, che converte dati da forma elettrica ad altra forma
Le funzioni di un modulo I/O sono:
-controllo e temporizzazione
-comunicazione con CPU
-comunicazione con periferiche
-buffering di dati (per non intasare ad esempio CPU con operazioni più lente delle periferiche)
-rilevazione errori

l. Descrivere il funzionamento di un modulo I/O.

-CPU interroga un modulo di I/O sullo stato di un dispositivo
-Modulo I/O verifica lo stato e restituisce risposta alla CPU
-Se dispositivo pronto, CPU invia a Modulo I/O richiesta trasferimento dati
-Modulo si interfaccia con periferiche per ottenere dati
-Ad operazione completata, Modulo I/O invia dati a CPU

l. Spiegare la differenza tra I/O memory mapped e I/O sparato

Sono due diverse modalità di indirizzamento quando CPU, memoria, e moduli di I/O condividono lo stesso bus.
-in I/O memory mapped memoria e periferiche condividono lo stesso bus
-operazioni di I/O sono quindi analoghe alle operazioni sulla memoria
-il vantaggio e che non c'è necessità di comandi speciali, ma vengono utilizzate le stesse istruzioni come per la memoria, c'è quindi un'ampia varietà di comandi
-uno svantaggio è che non è compatibile con architetture a bus multipli (dispositivi I/O non possono rispondere a bus non connessi).
Nell'I/O separato, troviamo spazi di indirizzamento separati tra memoria e periferiche.
Nel bus, è necessario l'utilizzo di un'ulteriore linea comando che specifica il tipo di indirizzo.
Necessita inoltre di comandi speciali per le operazioni di I/O (quindi numero comandi limitato).