**Rezolvare :**

**Biletul 1**

**1.Explicați notiunea " Memorie virtuala".**

Mecanismul de memorie virtuală este folosit de către nucleul sistemului de operare pentru a implementa o politică eficientă de gestiune a memoriei. Astfel, cu toate că aplicaţiile folosesc în mod curent memoria virtuală, ele nu fac acest lucru în mod explicit. Există însă câteva cazuri în care aplicaţiile folosesc memoria virtuală în mod explicit. Sistemul de operare oferă primitive de mapare a fişierelor, a memoriei sau a dispozitivelor în spaţiul de adresă al unui proces.

● Maparea fişierelor în memorie este folosită în unele sisteme de operare pentru a implementa mecanisme de memorie partajată. De asemenea, acest mecanism face posibilă implementarea paginării la cerere şi a bibliotecilor partajate.

● Maparea memoriei în spaţiul de adresă este folositoare atunci când un proces doreşte să aloce o cantitate mare de memorie.

● Maparea dispozitivelor este folositoare atunci când un proces doreşte să folosească direct memoria unui dispozitiv cum ar fi placa video.

Ideea este de a face distinctia intre capacitatea de adresare fizica a sistemului, care defineste un spatiu de memorie denumit **memorie virtuala**, si capacitatea de memorie instalata in sistem, denumita **memorie fizica** sau **reala**. In prezent, denumirea de memorie virtuala se foloseste cu doua acceptiuni: spatiu virtual care poate fi accesat (initial) sau zona de memorie plasata nu in memoria principala, ci pe un dispozitiv de memorare secundar, cel mai frecvent pe hard disk.

Acceptiunile termenului de **memorie virtuala** sunt urmatoarele:

1.    prima definitie de la memoria fizica;

2.    spatiu de memorie pe care il creeaza sistemul de operare si care contine un model al memoriei si al adreselor de memorie pe care le are programul la dispozitie (poate fi mai mare decat 1);

3.    suportul fizic, de obicei hard disk-ul, pe care sistemul de operare salveaza zone din memoria totala disponibila programului pentru a asigura spatii libere in zona de memorie fizica, destinate zonelor de cod sau de date cu care programul lucreaza la momentul respectiv.

**2.Instrucţiunea mov word ptr [bx],0:**

* 1. încarcă registrul **bx** cu valoarea **0**
  2. încarcă în locaţia adresată de **bx** valoarea **0** pe un octet
  3. **încarcă în locaţia adresată de bx valoarea 0 pe un cuvânt**

**3.Definiți modurile de funcționare a magistralei USB**

**Universal Serial Bus** (**USB**) este o specificațiepentru cabluri, conectori și protocoale de comunicații folosite pentru conectarea, comunicarea și alimentarea cu energie electrică între diverse dispozitive.

Magistrala USB reprezintã soluţia oferitã comunicaţiilor seriale de noua generaţie de calculatoare PC. Este o interfaţã serialã rapidã, bidirecţionalã, ieftinã şi uşor de folosit. USB a fost creatã ca un standard industrial, o extensie a arhitecturii PC orientatã spre armonizarea cu standardele de comunicaţie din telefonie, ceea ce este numit CTI (Computer Telephony Integration). Acest aspect este considerat fundamental din punct de vedere al aplicaţiilor generaţiei urmãtoare.

Magistrala USB defineşte trei categorii de dispozitive fizice:

• gazda USB (USB Host)

• funcţii USB (USB function)

• distribuitoare USB (USB Hub)

 Există două feluri de a trimite informații printr-un cablu sau chiar de la aplicație la aplicație: [serială](https://ro.wikipedia.org/wiki/Transmisie_serial%C4%83_de_date) și [paralelă](https://ro.wikipedia.org/wiki/Transmisie_de_date_paralel%C4%83); *serial* înseamnă [bit](https://ro.wikipedia.org/wiki/Bit) cu [bit](https://ro.wikipedia.org/wiki/Bit) pe un singur canal de comunicație și *paralel* înseamnă tot [bit](https://ro.wikipedia.org/wiki/Bit) cu [bit](https://ro.wikipedia.org/wiki/Bit) dar in paralel pe mai multe canale de comunicație.

**Tipurile de transfer de date**

Arhitectura USB permite patru tipuri de transferuri de date:

Transferurile *de control*sunt necesare la initierea dispozitivului.

Transferurile *de întrerupere*se utilizează pentru date cu volum redus. Transferul acestor date poate fi  solicitat de un dispozitiv in orice moment, iar rata de transfer pe magistrala USB nu poate fi mai redusă decat  cea specificată de dispozitiv. Datele pentru care se utilizează transferurile de intrerupere constau din notificarea  unor evenimente, din caractere sau coordonate care sunt organizate pe unul sau mai mulţi octeţi. Un exemplu il  reprezintă coordonatele de la un dispozitiv indicator (Ex. mouse). Datele interactive pot avea anumite limite ale  timpului de răspuns care trebuie asigurate de magistrala USB.

Transferurile *de date voluminoase*(“*bulk*”) se utilizează cu periferice cum sunt memorii de masă, imprimante sau scanere. Aceste date sunt secvenţiale. Fiabilitatea transferurilor este asigurată la nivel hardware  prin utilizarea unui cod detector de erori şi reluarea unui transfer cu erori de un număr de ori. Rata de transfer in  cazul acestor transferuri poate varia in funcţie de alte activităţi de pe magistrală.

Transferurile *izocrone*(*isos*– egal, *chronos*– timp) se utilizează pentru datele care trebuie furnizate cu o  anumită rată de transfer constantă şi a căror sincronizare trebuie garantată.

Info suplimentară : <http://vega.unitbv.ro/~ogrutan/Interfete%20si%20protocoale/Curs/9-Magistrala%20USB.pdf>

**Biletul 2**

**1.Menţionaţi aspecte ale functionarii dispozitivelor USB combinate .**

Magistrala USB reprezintã soluţia oferitã comunicaţiilor seriale de noua generaţie de calculatoare PC. Este o interfaţã serialã rapidã, bidirecţionalã, ieftinã şi uşor de folosit. USB a fost creatã ca un standard industrial, o extensie a arhitecturii PC orientatã spre armonizarea cu standardele de comunicaţie din telefonie, ceea ce este numit CTI (Computer Telephony Integration).

Un aspect care s-a avut în vedere la elaborarea magistralei USB a fost asigurarea unei rate de transfer mai ridicate decât ratele de transfer permise de porturile seriale şi paralele. De asemenea, s-a urmărit ca perifericele să poată fi adăugate în mod simplu la calculator, fără deschiderea carcasei acestuia, fără oprirea tensiunii de alimentare şi fără reîncărcarea sistemului de operare.

**2.Instrucţiunea CLC:**

* 1. activează sistemul de întreruperi
  2. dezactivează sistemul de întreruperi
  3. **pune carry pe 0**
  4. pune carry pe 1

**3.Destinația puntilor in arhitectura clasica a calculatoarelor personale.**

PCI (Peripherial Component Interconnect Bus) Este o alternativa la VLB, dar presupune intercalarea unor punti (bridge) intre magistrala procesorului si magistrala de I/E

De obicei, un chipset este format din două punți:

Puntea de nord este un controler de sistem, care este unul dintre elementele chipset-ului plăcii de bază și este responsabil pentru funcționarea procesorului central cu RAM, adaptor video și puntea de sud.

Prerogativa puntei de sud este operațiunile „lente”. El este responsabil pentru funcționarea interfețelor SATA, IDE, LAN, USB, BIOS, audio și puntea de nord. Puntea de sud are și un al doilea nume - controler I / OS

**Biletul 3**

**1.Descrieți componentele funcționale ale unui calculator" Von Neumann"(hardware).**

Arhitectura Von Neumann descrie un calculator cu 4 module : *Unitatea aritmetica logica, unitatea de control, memoria contrala is dispositive de intrare si iesire.*

*Unitatea aritmetica logica* este creierul calculatorului, care poate efectua mai multe tipuri de operatii: operatii aritmetica, de comparatie de manevrarea a datelor si de influentare a ordinei si fluxurilor instuctiilor

*Dispozitivele de Intrarea si Iesire (I/O)* sunt dispositive prin care calculatorul preia informatia si raporteaza inapoi rezultatele.

*Unitatea de control* este un modul central care comanda si leaga toate modulele intre ele. Rolul fiind sa citeasca instructiile si datele din memorie sau de la dispositive de Intrare si Iesire, sa ofere *Unitatii aritmetice logice* date de intrare corecte conform cu instructiunea si sa comande ce operatii sa efectueze asupra intrarilor, precum sis a trimita rezultatele.

**2.Instrucţiunea out 71h,al**

* 1. încarcă în registrul al valoarea 71h
  2. citeşte în registrul al un octet de la portul 71h
  3. **scrie valoarea din registrul al la portul 71h**

**3.Conceptul "echipamente periferice" pentru calculatoarele PC.**

Orice calculator PC interactionea cu utilizatorul prin intermediul unor dispositive de intrare si iesire. Datele se introduce in PC prin intermediul porturilor care sunt specific fiecarui tip de dispositive si apoi sunt transformate intr-un format numeric.

Intre dispozitivele periferice sunt:

-dispositive periferice de intrare: Tastatura, Mouse, Scanner, Camerele de luat vederi.

-dispositive periferice de iesire : Monitor, Imprimanta, Boxe, Videoproiector.

-Dispositive periferice de intrarea-iesire : Modem, Hard Disck.

**Biletul 4**

1. **Caracterizați magistrala PCI.**

Magistrala PCI este o magistrala paralela de date pentru atasarea dispozitivelor hardware la placa de baza periferice dintr-un calculator. Această magistrală de extensie a fost destinată procesoarelor Pentium şi prima versiune a apărut în anul 1992. Fiecare dispozitiv conectat la magistrala PCI poate funcţiona fie ca slave, fie ca master/slave. Un dispozitiv ţintă nu poate iniţia transferuri pe magistrală. Standardul magistralei PCI definește: - parametrii fizici (de exemplu, conectori și rutare linie de semnal); - parametrii electrici (de ex. tensiune); - model logic (de exemplu, tipuri de cicluri de magistrală, adresare de magistrală).

1. **Instrucţiunea JNZ et realizează saltul la et dacă:**
   1. flagul Zero este 1
   2. **flagul Zero este 0**
   3. flagul Sign este 1
   4. flagul Sign este 0
2. **Explicați notiunea GUI (Graphic User Interface).**

Interfața grafică (Graphical User Interface sau GUI) este o interfață cu utilizatorul bazată pe un sistem de afișaj ce utilizează elemente grafice. Interfața grafică este numit sistemul de afișaj grafic-vizual pe un ecran, situat funcțional între utilizator și dispozitive electronice cum ar fi computere, dispozitive personale de tip hand-held, aparate electrocasnice etc. Pentru a prezenta toate informațile și acțiunile disponibile, un GUI oferă pictograme și indicatori vizuali, în contrast cu interfețele bazate pe text, care oferă doar nume de comenzi sau navigația text

**Biletul 5**

1. **Cardul Grafic si Arhitectura SIMD.**

Un procesor grafic este un procesor pentru computere și jocuri console , care este specializat și optimizate pentru calculul grafice și smartphone - uri . În plus, transmite grafica calculată la unul sau mai multe afișaje. Obișnuia să aibă Cardurile grafice nu aveau capacități de calcul proprii și erau carduri de ieșire pure.

Această categorie de arhitecturi cuprinde sistemele de calcul compuse din mai multe unităţi de execuţie identice aflate sub comanda unei singure unităţi de control.

Unitatea de control transmite acelaşi flux de instrucţiuni, simultan, tuturor unităţilor execuţie. Toate unităţile de execuţie execută simultan aceeaşi instrucţiune asupra datelor din memoria proprie (există sisteme ce au şi o memorie partajată pentru comunicaţii). Unitatea de control trebuie să permită tuturor elementelor de procesare să-şi termine instrucţiunea curentă înainte de iniţierea unei noi instrucţiuni, astfel că execuţia instrucţiunilor trebuie sincronizată între toate unităţile de execuţie. Ca şi ordin de mărime numărul procesoarelor implicate într-o structură SIMD este de câteva mii.

1. **Instrucţiunea in al,71h:**
   1. încarcă în registrul **al** valoarea 71h
   2. **citeşte în registrul al un octet de la portul 71h**
   3. scrie valoarea din registrul **al** la portul 71h
2. **Explicați noţiunea "Memorie cu acces Direct".**

Accesul direct la memorie (DMA - Direct Memory Access) este folosit pentru transferurile rapide dintre dispozitivele de intrare/ieşire şi memorie fară intervenţia CPU. Accesul direct la memorie este folosit în driver-ele de floppy disk şi de hard disk, dar poate fi folosit şi la transferuri de la alte dispozitive de intrare sau ieşire (Sound Blaster, CD-ROM, plăci de reţea, etc) şi memorie. Transferuri DMA se pot face în ambele direcţii între memorie şi porturile de intrare/ieşire precum şi între două zone de memorie.

**Biletul 6**

1. **Explicati procesele de transmisie de date "sincronice" si "asincronice".**

Exista doua tipuri de transmisie: - sincrona - asincrona Este necesar un mecanism pentru realizarea sincronizarii operatiilor de transfer intre 2 calculatoare, astfel incat informatiile receptionate sa fie identice cu cele emise pe linia seriala. Sincronizarea se poate face fie prin sincronizarea semnalelor de tact ale celor doua parti (suport hard), fie prin continutul mesajului transmis (suport soft).

***Transmisiile sincrone*** necesita o operatie de sincronizare initiala si promisiunea de stabilitate a ceasurilor la cele 2 parti. Datele sunt transmise continuu, iar pentru perioadele de pauza se transmite codul unui caracter special (idle).

***Transmisiile asincrone*** realizeaza sincronizarea la nivelul fiecarui cod (octet), atasand acestuia biti suplimentari. Intre transmisia a 2 octeti linia este pe “1” (asigura detectarea purtatoarei), primul bit va fi bitul de START ( = 0 ), urmeaza un numar predefinit de biti de date si 1 sau 2 biti de STOP ( =1 ) .

Sincronic -În transmisia sincronă, fluxul de date care urmează să fie transferat este codificat ca tensiuni fluctuante într-un fir și un impuls periodic de tensiune este pus pe un alt fir (numit adesea "ceas" sau "strobe").Expeditorul și receptorul utilizează același semnal de ceas.

Asincrona-Intr-o formă de transmisie asincronă, există doar un fir / semnal care transporta transmisia. Expeditorul furnizează un semnal de sincronizare receptorului înainte de a începe transferul fiecărui mesaj.

Asincrone și sincron sunt două metode diferite de sincronizare a transmisiei. Diferența majoră dintre acestea constă în metodele lor de transmitere, adică transmisiile sincrone sunt sincronizate de un ceas extern; în timp ce transmisiile asincrone sunt sincronizate prin semnale speciale de-a lungul mediului de transmisie.

1. **Instrucţiunea corespunzătoare etichetei et din LOOP et se află:**
   1. **înaintea instrucțiunii LOOP**
   2. după Instrucţiunea LOOP
   3. nu se afla în programul respectiv
2. **Organizarea datelor pe unitati de disc tip Winchester.**

Datele sunt reprezentate in sistemul binar si sunt organizate in fisiere de diferite tipuri, care sunt grupate dupa anumite criterii in directoare(dosare).Un director poate contine mai multe directoare si/sau fisiere sau poate fi gol.Dupa formatarea HDD-lui se va crea un director care este numit directorul radacina care va fi singurul imediat dupa formatare. Toate celelalte directoare(sau dosare) si fisiere se vor crea ulterior in directorul radacina. Directoarele sunt organizate intr-o structura arborescenta.

**Biletul 7**

1.**Explicați notiunea "magistrală".**

***Magistrală-*** un set de conexiuni fizice(un grup de semnale electrice sau fire) între blocuri prin care informaţia care circulă are o semnificaţie prestabilită.

Magistrala este un ansamblu de conexiuni electrice prin care circulă informație de acelaşi tip având ca suport semnale electrice; în funcție de tipul informației, magistralele sunt de trei categorii: de date, de adrese şi de control. O caracteristică de bază este dimensiunea magistralei, adică numărul liniilor de conectare; avem astfel magistrale de 8 biți (cu 8 linii de conectare), magistrale de 16 biți (cu 16 linii de conectare), etc. Dimensiunea fiecărei magistrale este determinată de structura unității centrale şi determină la rândul său structura memoriei (numărul de biți pe locație) şi a porturilor de intrare / ieşire.

Prin cuvântul magistrală se înțelege de regulă şi ansamblul de circuite electronice (amplificatoare uni- sau bi-direcționale) care sporesc puterea semnalelor electrice (pentru fan-out mai mare) şi aduc nivelul (tensiunea) la valoarea standard.

Magistralele unidirecționale pot transmite informația într-un singur sens iar cele bidirecționale, în ambele sensuri (sensul de transmisie este controlat de unitatea centrală). Magistrala de adrese este unidirecțională (de la unitatea centrală spre sistem) iar magistralele de date şi de control sunt bidirecționale.

Magistrala paralelă poate fi subdivizată in trei categorii, după tipul informaţiei transferate: magistrale de adrese, de date, şi de control.

Destinaţia liniilor magistralei paralele

Magistrala asigură conectarea paralelă a dispozitivelor periferice şi asigură transferurile de informaţie între ele.

Arhitectura magistralei include următoarele componente:

- Linii pentru transferuri de date (magistrala de date);

- Linii pentru adresarea datelor (magistrala de adrese);

- Linii pentru controlul datelor (magistrala de control);

- Controllerul magistralei.

Controllerul magistralei efectuează controlul procesului de schimb de date, de semnale de control şi reprezintă un microcircuit sau se integrează în circuitele chipset-ului.

**2.Instrucţiunea adc dest, sursă realizează operaţia:**

* 1. sursă<--dest+ sursă
  2. dest<--dest+ sursă
  3. **dest<--dest+ sursă+carry**
  4. sursă<--dest+ sursă+carry

**3.Calculati volumul de memorie minimal necesar pentru a stoca o imagine statica in color (RGB) la rezolutie 1024 linii cu cate 1080 pixeli per linie.**

Dimensiune imagine 1024×1080 pixeli, 1.105.920 pixeli,1.106 Megapixeli, Raport de aspect 135:128 (1.055:1)

Dimensiune date 3,317,760 Bytes, 3.164 MB, 3 bytes per pixel

Pentru a calcula volumul de memorie minimal trebuia mai întâi să calculăm numărul de pixeli al imaginii 1024x1080=1.105.920, deoarece e o imagine statică trebuie să înmulțim cu 3 pentru a afla numărul de octeți , pentru fiecare culoare câte 1. 1.105.920x3=3.317.760 am aflat numărul de octeți, acum știm că un kilobyte este egal cu 1026 bytes. Deci pentru a putea converti împărțim numărul de octeți la 1024. 3.317.760%1024=3.240 kilobytes, nu ne oprim aici și calculăm cât va fi în megaocteți pentru a putea calcula este necesar de împărțit numărul de kilobytes la 1024. 3240%1024=3,16 megaocteți sau MB. Volumul de memorie minimal pentru a stoca această imagine este 3,16MB

**Biletul 8**

**1.Explicați principiul de funcționare a controllerului DMA**

Controllerul DMA (DMA-direct memory access, acces direct la memorie) este un dispozitiv ce asigură accesul direct la RAM a perifericelor, neutilizând registrii interni ai microprocesorului. Cererea de acces este formulată pe un terminal specializat denumit, de regulă, "BUSRQ" (cerere de magistrală). Are prioritatea maximă şi răspunsul aşteaptă doar terminarea ciclului maşină curent. Microprocesorul "îngheaţă" întreaga activitate internă; magistralele sale trec în "impedanţă înaltă" iar controlul transferului de informație este preluat de controllerul DMA. Acesta facilitează transferul informaţiei direct între memorie şi echipamente periferice.

In cazul în care controlerului DMA funcționeaza, la o frecventamaximă de 15 MHz, cu o memorie de 50 ns, rata maximă de transfer este de 15 MHz (67ns), deoarece controlerul DMA este mai lent decât memoria

In multe cazuri, controlerul DMA încetinește viteza sistemului atunci când au loc transferuri

**2.Cum sunt reprezentate în memorie caracterele?**

In categoria datelor alfanumerice intra caracterele alfabetice, caracterele numerice, semnele de punctuație şi caracterele de control. Codificarea acestora se face conform codului standard ASCII (American Standard Code for Information Interchange). Codul ASCII standard este un cod pe 7 biți (128 combinații distincte). Ca atare un caracter ASCII se reprezintă pe un octet in care bitul cel mai semnificativ este zero.Domeniul de valori este de la 0 la 127 (in hexazecimal de la 00h la 07fh).

Exemplu:

Caracterul 'a' are codul 61h, 'A' are codul 41h, '5' are codul 35h, '+' are codul 02Bh. Șirurile de caractere sunt reprezentate prin succesiuni de octeți ce cuprind codurile caracterelor respective. De exemplu șirul de caractere “Aa4+b” se reprezintă pe 5 octeți ce vor conține 41 61 34 2B 42h

**3.Explicati notiunile ”*parity chek, CRC(Cicl”.***

Parity check – procesul care asigură transmiterea exactă a datelor între noduri în timpul comunicării. Un bit de paritare este atașat la biții de date originali pentru a crea un număr de biți par sau impar, numărul de biți cu valoarea unu. Sursa transmite transmite apoi date printr-un link, iar biții sunt verificați la destinație. Datele sunt considerate corecte dacă numărul de biți se potrivește cu numărul transmis de la sursă.

CRC cicl – cod ciclic redundant, este un cod de detectare a erorilor utilizat în mod obișnuit în rețelele digitale și dispozitivele de stocare pentru a detecta modificările accidentale ale datelor brute. Blocurile de date care intră în aceste sisteme primesc o scurtă valoare de verificare, bazată pe restul unei diviziuni polinomiale a conținutului lor. La recuperare, calculul se repetă și, în cazul în care valorile verificării nu se potrivesc, pot fi luate măsuri corective împotriva corupției datelor. CRC-urile pot fi utilizate pentru corectarea erorilor

Biletul nr. 9

\*\*Pentru evaluarea 2(doi)la disciplina: Arhitectura calculatoarelor

**Grupele AI 191;SI191;TI1912,3,4,5;FI191 Facultatea C.I.M.**

1. **Explicati notiunea HUB pentru magistrala USB.**

Un hub reprezinta un multiplicator a conectoarelor. Hub-urile asigură puncte de conectare suplimentare a dispozitivelor la magistrală.Dispozitivele hub sunt elemente cheie în arhitectura plug-and-play a USB. Hub-urile sunt folosite pentru a simplifica conectica din punctul de vedere al utilizatorului şi totodatã determinã un sistem robust şi ieftin. Punctele de ataşare se numesc porturi. Existã un port spre gazdã (amonte - upstream port) şi mai multe porturi spre alte dispozitive (aval - downstream port). Un hub are douã componente: un repetor (Hub Repeator) şi un controler (Hub Controller). Repetorul este un comutator comandat prin protocol, care face legãtura între portul amonte cu unul din porturile aval. Controlerul conţine registre de interfaţã care fac posibilã comunicarea cu gazda, pentru configurare şi comandã.

1. **Instrucţiunea cmp *dest,* *sursă* realizează operaţia:**
   1. compara operanzii sursă si destinaţie prin aplicarea funcției logice "si"
   2. compara operanzii sursă si destinaţie prin aplicarea funcției logice "sau"
   3. compara operanzii sursă si destinaţie prin scădere.

**3. Tipuri de Magistrale**.

Conceptual, magistrala este un mediu comun de comunicatie intre componentele unui sistem de calcul; fizic este alcatuita dintr-un set de linii de semnal care faciliteaza transferul de date si realizeaza sincronizarea intre componentele sistemului. In decursul timpului au fost dezvoltate diferite standarde de magistrala, care au urmarit evolutia procesoarelor (a unitatilor centrale) si a necesitatilor de comunicatie ale acestora (viteza, mod de transfer, necesitati de sincronizare si control etc.). S-au dezvoltat, de asemenea, magistrale specializate pentru anumite tipuri de echipamente periferice (SCSI si EIDE pentru unitati de disc, console grafice, interfete de masura si control). Din acest punct de vedere. in momentul actual se pot distinge doua clase de magistrale:

·        **magistrale de sistem** – dezvoltate mai ales pentru conectarea unitatii centrale la celelalte componente de baza ale sistemului (de exemplu: MultiBus, ISA, EISA, PCI);

·        **magistrale specializate** – care incearca sa optimizeze transferul de date cu un anumit tip de echipamente periferice (VESA, AGP, SCSI, GPIB).

Magistralele pot fi clasificate in functie de mai multe **criterii**.

I. **modul de lucru** (in raport cu semnalul de tact):

·         **magistrale sincrone** – la care ciclurile de transfer sunt direct corelate cu semnalul de tact. Viteza de transfer este mai mare, insa rata de transfer a magistralei este limitata de frecventa tactului. Datorita vitezei limitate de propagare a semnalului electric, cresterea ratei ar duce la diferente de faza la capetele magistralei.

·         **magistrale asincrone***–*la care nu exista o legatura directa intre evolutia in timp a unui ciclu de transfer si tactul sistemului. Majoritatea magistralelor actuale lucreaza pe acest principiu (de exemplu, ISA, EISA, MultiBus).

II. **numarul de module master** conectate pe magistrala:

·         **magistrale unimaster***–* exista un singur modul master pe magistrala. Nu necesita mecanisme de arbitrare a magistralei (un **modul master** poate initia un ciclu de transfer, pe cand un **modul slave** poate fi comandat in timpul unui ciclu de transfer, neavand elementele necesare pentru a prelua controlul magistralei).

·         **magistrale multimaster***–*permit conectarea mai multor module master pe acelasi tronson de magistrala. Magistrala trebuie sa contina semnale de arbitrare si un protocol de transfer al controlului pe magistrala (de exemplu, MultiBus, VME).

III. **modul de realizare al transferului de date**:

·      **magistrale cu transfer prin cicluri**(**magistrale secventiale**)*–* regula de baza: ciclurile de transfer se desfasoara secvential, la un moment dat cel mult un ciclu de transfer fiind in curs de desfasurare. Majoritatea magistralelor folosesc acest principiu de transfer.

In functie de **directia de transfer** se disting sase cicluri semnificative:

¡      cicluri de citire (respectiv scriere) din/in memorie;

¡      cicluri de citire (respectiv scriere) de la/la un port intrare-iesire;

¡      cicluri de acces direct la memorie (citire/scriere).

·      **magistrale tranzactionale***–* transferul de date se efectueaza prin tranzactii. O tranzactie este divizata in mai multe faze. Mai multe tranzactii se pot desfasura simultan, cu conditia ca tranzactiile sa fie in faze diferite; aceasta restrictie provine din faptul ca fiecare faza a unei tranzactii foloseste un subset din multimea semnalelor magistralei. Teoretic, la aceste magistrale factorul de crestere a vitezei (in comparatie cu o magistrala secventiala) este egal cu numarul de faze in care se divide o tranzactie (de exemplu, magistrala procesorului Pentium).

IV. **numarul semnalelor utilizate in transferul de date**:

·      **magistrale seriale***–*se utilizeaza rar ca mijloc de comunicatie intre componentele de baza ale unui calculator (CPU, memorie, interfete de I/O etc.), datorita vitezei de transfer relativ scazute. Se folosesc, totusi, in anumite sisteme dedicate bazate pe microcontrolere, la care costul si dimensiunea redusa sunt parametri definitorii (de exemplu, magistrala USB).

·      **magistrale paralele** – cele mai raspandite.

In continuare se va face referire la **magistralele paralele**.

Proces verbal nr. 4 din 24.11.2020

Biletul nr. 10

\*\*Pentru evaluarea 2(doi)la disciplina: Arhitectura calculatoarelor

**Grupele AI 191;SI191;TI1912,3,4,5;FI191 Facultatea C.I.M.**

1. Definiţi noţiunea " MAGISTRALA "(BUS).

Magistrala este calea prin care pot circula datele în interiorul unui calculator, între două sau mai multe elemente ale acestuia. Calculatorul are o ierarhie a diferitelor magistrale deoarece magistralele mai lente sunt legate la magistrale mai rapide. Fiecare dispozitiv din sistem este conectat la una din magistrale, unele dintre ele functionînd ca punți între magistrale. Principalele magistrale ale unui PC sunt:

1. **Magistrala procesorului** care este cea mai rapidă și face parte din nucleul setului de cipuri și al plăcii de bază. Magistrala procesorului este calea de comunicare între CPU (unitatea centralăde prelucrare și cipurile cu care lucrează direct. Această magistrală este folosită pentru a transfera date între CPU și magistrala principală a sistemului sau între CPU și memoria cache externă. În sistemele Pentium II, magistrala procesorului funcționează la frecvența de 66 sau 100 MHz și are lățimea de bandă de 64 de biți.
2. **Magistrala AGP** (Accelerated Graphics Port) care este o magistrală de mare viteză de 66 MHz, pe 32 de biți destinată în special pentru o placă video.
3. **Magistrala PCI** (Pheripheral Component Interconnect) apare în sistem ca un ansamblu de conectoare pe 32 de biți pentru adaptoare SCSI, plăci de rețea, plăci video.

**Magistrala ISA** (Industry Standard Architecture) este o magistrală foarte lentă cu o frecvență de 8 MHz, pe 16 biți, dar care se menține pentru legătura cu diferite periferice mai lente cum ar fi modemul și placa de sunet

1. Cum sunt reprezentate numerele reale:
   1. în mărime şi semn
   2. în format BCD
   3. în format virgula mobila
2. Explicați mecanismul functionarii memoriei "Cash".

 un cache este o arie temporară de stocare unde datele utilizate în mod frecvent pot fi depozitate pentru un acces rapid la acestea. Odată ce datele sunt stocate în cache, în viitor vor fi luate de aici și utilizate decât să se încerce readucerea datelor originale sau recalcularea acestora, astfel încât timpul mediu de acces este mai mic.

 Memoria cache, sau RAM cache-ul este memorie de tip static [RAM](https://ro.wikipedia.org/wiki/Memorie_cu_acces_aleator). Ca orice SRAM, are o viteză și un cost mult mai ridicate decât RAM-ul dinamic (DRAM). Având în vedere că programele accesează memoriile de date sau instrucțiuni în repetate rânduri, s-a observat că prin păstrarea a cât mai multe dintre aceste informații pe SRAM, întregul sistem funcționează mai rapid.

Memoriile de tip cache s-au dovedit a fi extrem de folositoare în multe domenii ale informaticii pentru că modelele accesului la memorie în [programele aplicație](https://ro.wikipedia.org/w/index.php?title=Programul_aplica%C8%9Bie&action=edit&redlink=1) obișnuite sunt de tip [localitate a referinței](https://ro.wikipedia.org/w/index.php?title=Localitatea_referin%C8%9Bei&action=edit&redlink=1). Există câteva feluri de localitate, dar acest articol prezintă cazul în care datele accesate sunt foarte apropiate în domeniul timp (fenomenul de [localitate temporală](https://ro.wikipedia.org/w/index.php?title=Localitatea_memoriei&action=edit&redlink=1)). Datele ar putea fi sau nu localizate fizic aproape una de cealaltă (localitate spațială).

Biletul nr. 11

\*\*Pentru evaluarea 2(doi)la disciplina: Arhitectura calculatoarelor

**Grupele AI 191;SI191;TI1912,3,4,5;FI191 Facultatea C.I.M.**

1. Explicati notiunea "compound device" pe magistrala USB.

(compound device...dispozitive care combina si imbina citeva functii si cabluri)

Ca dispozitive (Device) USB pot fi hub-uri (distribuitoare), funcţii sau dispozitive ce îmbină şi hub- uri şi funcţii (compound device, figura 7.17). Hub-urile asigură puncte de conectare suplimentare a dispozitivelor la magistrală. Funcţiile USB sunt nişte sisteme, de exemplu mouse ce susţine USB. Dispozitive care combină câteva funcţii, (compound device, figura mai jos) de exemplu, keyboard cu trackball încorporat.

Elementele principale ale unui sistem care utilizează magistrala USB sunt dispozitivele USB, cablurile USB şi programele de sistem. Dispozitivele de pe magistrala USB sunt conectate fizic la calculatorul gazdă utilizând o topologie sub formă de stea , după cum se ilustrează în Figura 7.17

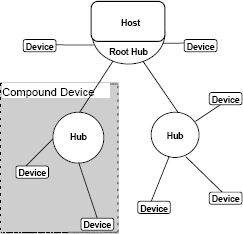


Figura 7.17 – Topologia fizică

1. Instrucţiunea out **71h,al**
   1. încarcă în registrul **al** valoarea 71h
   2. citeşte în registrul **al** un octet de la portul 71h
   3. scrie valoarea din registrul **al** la portul 71h
2. Definiţi noţiunea “interfaţă”.

**Interfaţa** – componenta ce asigură o conexiune între două elemente pentru ca acestea să poată lucra împreună. Interfeţele pot fi de mai multe tipuri:

Ø *Paralele* – transmit simultan un byte (8 biţi de informaţie); se folosesc, în general, pentru conectarea imprimantelor;

Ø *Seriale*– biţii unui byte se transmit pe rând, sub forma unor impulsuri; în general, sunt utilizate la conectarea la un modem sau la alt calculator;

Ø *USB (Universal Serial Bus)*– pentru conectarea oricărui periferic.

Biletul nr. 12

\*\*Pentru evaluarea 2(doi)la disciplina: Arhitectura calculatoarelor

**Grupele AI 191;SI191;TI1912,3,4,5;FI191 Facultatea C.I.M.**

1. Nivele de privilegii ale programelor .

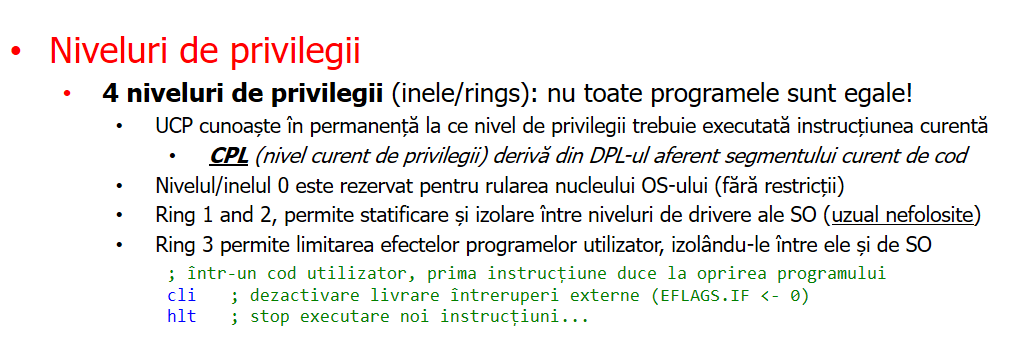
Niveluri de privilegii

Exista 4 niveluri de privilegii:

UCP cunoste in permanenta la ce nivel de privilegii trebuie executata instructiunea curenta

Niveluri:

* CPL (nivel curent de privilegii) deriva din DPL-ul aferent segmentului curent de cod
* Nivelul/inelul 0 este rezervat pentru rularea nucleului OS-ului (fara restrictii)
* Nivelul 1 si 2, permite statificare si izolare intre niveluri de drivere ale SO(uzual nefolosite)
* Nivelul 3 permite limitarea efectelor programelor utilizator, izolandu-le intre ele si de SO



1. Instrucţiunea STC:
   1. activează sistemul de întreruperi
   2. dezactivează sistemul de întreruperi
   3. pune carry pe 0
   4. pune carry pe 1
2. Descrieti arhitectura Hyper Transport

Tehnologia (arhitectura) HyperTransport (HT) a fost proiectată ca alternativă arhitecturii bazate pe magistrale şi punţi. Principala inovaţie – înlocuirea conectării prin intermediul magistralelor a dispozitivelor periferice prin conectarea punct-la-punct (point-to-point) ce presupune transferuri de date concomitent în ambele direcţii între 2 noduri (Prin această tehnologie se majorează considerabil viteza de transfer a datelor.Puntea de bază (host bridge) asigură legătura cu nucleul sistemului – microprocesorul şi memoria. Controllerele perifericelor, ce necesită rate înalte de transfer de date sunt realizate în formă de tuneluri HT (HT Tunnel). Caracteristicile principale ale tunelului sunt:

− Interfaţa HyperTransport pe 16 biţi (Side A) oferă o rată maximală de transfer (lăţime de bandă) – până la 6,4 GB/s; − Interfaţa HyperTransport pe 8 biţi (Side B) oferă o rată maximală de transfer – până la 1,6 GB/s.

**12 martie 2021 Examinator: Postovan Dumitru** 

Biletul nr. 13

\*\*Pentru evaluarea 2(doi)la disciplina: Arhitectura calculatoarelor

**Grupele AI 191;SI191;TI1912,3,4,5;FI191 Facultatea C.I.M.**

1. **Caracterizați grupa de calculatoare – mainframe.**

 Calculatoarele mainframe sunt calculatoare ce pot exploata volume imense de date şi pot suporta lucrul a mii de utilizatori simultan. Un calculator mainframe se distinge mai ales prin capacitatea de stocare şi memoria internă. El poate rula ani întregi fără întrerupere, Unele calculatoare pot rula mai multe sisteme de operare simultan, operând astfel ca o mulţime de “maşini virtuale”. Preţul unui astfel de calculator este de ordinul sutelor de mii de dolari. Este solicitat de companiile care vehiculează şi prelucrează un volum foarte mare de informaţie 19 Clasificarea lui Flynn. Arhitectura SISD. Instrucţiunile sunt executate secvenţial, însă pot exista suprapuneri între acestea dacă este implementat conceptul de bandă de asamblare (pipeline) – majoritatea sistemelor SISD actuale utilizează conceptul de bandă de asamblare. Calculatoarele SISD pot avea mai multe unităţi funcţionale (ex: coprocesor matematic, procesor grafic, procesor de intrare/ieşire, etc.), însă acestea sunt văzute ca o singură unitate de execuţie. 20 Clasificarea lui Flynn. Arhitectura SIMD. Această categorie de arhitecturi cuprinde sistemele de calcul compuse din mai multe unităţi de execuţie identice aflate sub comanda unei singure unităţi de control. Unitatea de control transmite acelaşi flux de instrucţiuni, simultan, tuturor unităţilor de execuţie. Toate unităţile de execuţie execută simultan aceeaşi instrucţiune asupra datelor din memoria proprie (există sisteme ce au şi o memorie partajată pentru comunicaţii). Unitatea de control trebuie să permită tuturor elementelor de procesare să-şi termine instrucţiunea curentă înainte de iniţierea unei noi instrucţiuni, astfel că execuţia instrucţiunilor trebuie sincronizată între toate unităţile de execuţie. Ca şi ordin de mărime numărul procesoarelor implicate într-o structură SIMD este de câteva mii. 21 Clasificarea lui Flynn. Arhitectura MISD. Arhitecturile MISD au mai multe elemente de procesare, fiecare executând un set diferit de instrucţiuni asupra unui singur flux de date. Acest lucru este realizabil în două moduri: acelaşi element din fluxul de date este prelucrat de toate procesoarele, fiecare executând propriile operaţii asupra respectivei date; un element din fluxul de date este prelucrat de primul procesor, rezultatul obţinut este pasat mai departe celui de-al doilea procesor ş.a.m.d., formându-se astfel o macro-bandă de asamblare.

1. Instrucţiunea **mov word ptr [bx],0**:
   1. încarcă registrul **bx** cu valoarea **0**
   2. încarcă în locaţia adresată de **bx** valoarea **0** pe un octet
   3. încarcă în locaţia adresată de **bx** valoarea **0** pe un cuvânt
2. **Explicați conceptul - memorie virtuala**

Memoria virtuală este o zonă de stocare temporară la care se apelează ori de câte ori un program necesită mai multă memorie RAM decât cea care se găseşte instalată într-un calculator.

Pentru ca memoria RAM să nu impiedice rularea unor programe atunci când aceastea au nevoie de mai mult spatiu de stocare,sistemul de operare foloseste spaţiul de pe hard disk ca o extensie a memoriei ram. Chiar dacă un calculator este echipat în realitate cu doar 64 de megabiţi de RAM, prin folosirea memoriei virtuale un program poate avea la dispoziţie până la 4 gigabaiţi de memorie. (in cazul unui sistem de operare pe 32 de biti)

În calcul , memoria virtuală sau stocarea virtuală este o tehnică de gestionare a memoriei care oferă o „abstractizare idealizată a resurselor de stocare care sunt de fapt disponibile pe o anumită mașină” care „creează iluzia utilizatorilor unei memorii (principale) foarte mari”.

Principalele avantaje ale memoriei virtuale includ eliberarea aplicațiilor de a gestiona un spațiu de memorie partajat, capacitatea de a partaja memoria utilizată de biblioteci între procese, securitatea sporită datorită izolării memoriei și posibilitatea de a utiliza conceptual mai multă memorie decât ar putea fi disponibilă fizic, folosind tehnica paginării sau segmentării .

Biletul nr. 14

\*\*Pentru evaluarea 2(doi)la disciplina: Arhitectura calculatoarelor

**Grupele AI 191;SI191;TI1912,3,4,5;FI191 Facultatea C.I.M.**

1. **Explicati notiunea "Magistrale Paralele".**

În functie de numărul liniilor utilizate pentru transferul de date, magistralele sunt de două tipuri: paralele şi seriale. Cele seriale se utilizează în principal pentru comunicatie între sisteme de calcul aflate la distantă (retele de calculatoare).

O magistrală paralelă se compune din următoarele tipuri de semnale:

semnale de date - utilizate pentru transfer de date în ambele sensuri; la un moment dat o singură unitate poate să emită pe liniile de date; numărul liniilor de date ( 8, 16, 32, 64) determină dimensiunea cuvântului de date ce poate fi transferat şi viteza de transfer;

semnale de adresă - având rolul de adresare a modulului destinatie sau sursă; numărul liniilor de adresă determină dimensiunea spatiului de adresare al magistralei;

semnale de comandă - specifică directia de transfer (dinspre procesor sau către procesor) şi tipul componentei adresate (modul de memorie, dispozitiv de intrare/ieşire, controler de întreruperi etc.).

semnale de control - stabilesc conditiile de transfer pe magistrală;

semnale de întrerupere - semnalizează aparitia unor evenimente interne sau externe şi determină întreruperea executiei programului curent;

semnale de tact

- au rol de sincronizare şi permit generarea unor semnale de frecventă programabilă;

semnale de alimentare :- utilizate ca tensiuni de alimentare pentru componentele sitemului; semnale de control acces –utilizate pentru arbitraj şi pentru controlul accesului la magistrală, în cazul magistralelor multimaster.

1. Instrucţiunea **JNC et** realizează saltul la **et** dacă:
   1. flagul Zero este 1
   2. flagul Zero este 0
   3. flagul Carry este 1
   4. flagul Carry este 0
2. **Caracterizați perifericele de tip Winchester.**

**Dispozitive de memorie externă**

Dispozitivele de memorie externă permit stocarea unor cantităţi mari de informaţii pe termen lung. În funcţie de tehnologia utilizată pentru stocarea informaţiilor acestea se clasifică în:   
- magnetice;   
- optice;   
- magnetico-optice.

1. **Dischete (FD – floppy-disc)** – suporturi de memorie magnetică. Au forma pătratică cu latura de 3,5", carcasă din plastic, pot stoca 1,44Mb (capacitate mică), viteza de acces la informaţii este mică dar sunt portabile şi ieftine. Pentru a putea utiliza o dischetă aceasta trebuie formatată. Operaţia de formatare stabileşte modul de memorare a informaţiilor pe dischetă. Citirea şi scrierea informaţiilor de pe dischete se realizează cu ajutorul unităţilor de floppy-disc.

2. Hard-disk (HDD – disc dur) – este un dispozitiv magnetic de stocare, este principala formă de stocare externă a informaţiilor, are o capacitate mare de stocare (de ordinul zecilor de Gb), viteza de acces la informaţii este mare (10-15 milisecunde), viteza de rotaţie este mare (până la 10000 rotaţii pe minut), sunt costisitoare, sunt puţin portabile. Există şi HDD-uri portabile incluse într-o carcasă de plastic dar necesită o unitate specială.

3. Discurile optice – sunt: - CD-ROM (Compact Disc Read Only Memory) – capacitate de stocare 630Mb - 1Gb; - CD-RW (CD ReWritable) – CD reinscriptibil); - DVD (Digital Versatile Disk) - capacitate de stocare mai mare decât a CD-urilor 4,7-17,08 Gb, viteza de acces la informaţii este mai mare decât viteza la CD. Fiecare dintre aceste dispozitive necesită unităţi speciale pentru citire respectiv scriere.

4. Discurile magnetico-optice – combină tehnologia discurilor magnetice cu cea CD-ROM. Pot fi citite şi scrise, sunt portabile, au capacitate mare de stocare (200Mb), viteza de acces la informaţii este mare.

5. Banda magnetică – este similară benzilor muzicale de casetofon. Sunt utilizate pentru stocarea pe termen lung a informaţiilor (arhive, copii etc.).

Biletul nr. 15

\*\*Pentru evaluarea 2(doi)la disciplina: Arhitectura calculatoarelor

**Grupele AI 191;SI191;TI1912,3,4,5;FI191 Facultatea C.I.M.**

1. **Explicați notiunea "magistrală".**

Magistrala este calea prin care pot circula datele în interiorul unui calculator, între două sau mai multe elemente ale acestuia. Calculatorul are o ierarhie a diferitelor magistrale deoarece magistralele mai lente sunt legate la magistrale mai rapide. Fiecare dispozitiv din sistem este conectat la una din magistrale, unele dintre ele functionînd ca punți între magistrale. Principalele magistrale ale unui PC sunt:

1. **Magistrala procesorului** care este cea mai rapidă și face parte din nucleul setului de cipuri și al plăcii de bază. Magistrala procesorului este calea de comunicare între CPU (unitatea centralăde prelucrare și cipurile cu care lucrează direct. Această magistrală este folosită pentru a transfera date între CPU și magistrala principală a sistemului sau între CPU și memoria cache externă. În sistemele Pentium II, magistrala procesorului funcționează la frecvența de 66 sau 100 MHz și are lățimea de bandă de 64 de biți.
2. **Magistrala AGP** (Accelerated Graphics Port) care este o magistrală de mare viteză de 66 MHz, pe 32 de biți destinată în special pentru o placă video.
3. **Magistrala PCI** (Pheripheral Component Interconnect) apare în sistem ca un ansamblu de conectoare pe 32 de biți pentru adaptoare SCSI, plăci de rețea, plăci video.

**Magistrala ISA** (Industry Standard Architecture) este o magistrală foarte lentă cu o frecvență de 8 MHz, pe 16 biți, dar care se menține pentru legătura cu diferite periferice mai lente cum ar fi modemul și placa de sunet

1. **Implementaţi Instrucţiunea xchg bx,cx folosind instrucţiuni de tip mov**

mov ax,cx

mov cx,bx

mov bx,ax

1. Structuri de date pe unitati de disk tip Winchester.

Mecanismele tehnologiei unității de disc reprezintă doar jumătate din poveste; țiglă altă jumătate este modul în care datele sunt structurate pe disc. Nu există nicio modalitate de a planifica configurații de stocare optime fără a înțelege cum sunt structurate datele pe suprafața platourilor unității de disc. Există următoarele structuri de date utilizate în unitățile de disc:

* Piste, sectoare și cilindri
* Partiții de disc
* Adresarea blocului logic
* Geometria unităților de disc și înregistrarea pe biți zonali

Piste, sectoare și cilindri

Platourile de discuri sunt formatate într-un sistem de cercuri concentrice, sau inele, numite piese. În cadrul fiecărei piste sunt sectoare, care subdivizează cercul într-un sistem de arce, fiecare formatat pentru a conține aceeași cantitate de date - de obicei 512 octeți. A fost odată, dimensiunea blocului sistemelor de fișiere a fost cuplată cu dimensiunea sectorială a unui disc. Astăzi dimensiunea blocului unui sistem de fișiere poate varia considerabil, dar este de obicei un multiplu de 512 octeți.

Cilindrii sunt sistemul de piste identice pe mai multe platouri din unitate. Brațele multiple ale unei unități se mișcă împreună în pas, poziționând capetele în aceeași locație relativă pe toate platourile simultan.

Partiții de disc

Partițiile de disc împart capacitatea unităților de disc fizice în containere logice. O unitate de disc poate avea una sau mai multe partiții, oferind utilizatorilor o modalitate de a crea flexibil diferite discuri virtuale care pot fi utilizate în scopuri diferite

.

De exemplu, un sistem ar putea avea partiții diferite pentru a rezerva capacitatea de stocare pentru diferiți utilizatori ai sistemului sau pentru aplicații diferite. Un motiv comun pentru utilizarea mai multor partiții este stocarea datelor pentru sistemele de operare sau sistemele de fișiere. Mașinile care sunt capabile să ruleze două sisteme de operare diferite, cum ar fi Linux și Windows, ar putea avea datele respective pe partiții de disc diferite.

Partițiile de disc sunt create ca o colecție contiguă de piese și cilindri. Vizual, vă puteți imagina partiții care arată ca inelele concentrice ale unei ținte de tir cu arcul, cu ochiul taurului înlocuind cu fusul motorului discului. Partițiile sunt stabilite începând de la marginea exterioară a platourilor și lucrând spre centru. De exemplu, dacă un disc are trei partiții, numerotate 0, 1 și 2, partiția 0 ar fi în exterior și partiția 2 ar fi cea mai apropiată de centru.

Adresarea blocului logic

În timp ce sistemul intern al cilindrilor, pistelor și sectoarelor este interesant, de asemenea, nu mai este folosit prea mult de sistemele și subsistemele care utilizează unități de disc. Adresele cilindrice, de urmărire și de sector au fost înlocuite cu o metodă numită adresare de blocare logică (LBA), care face discurile mult mai ușor de lucrat cu prezentarea unui singur spațiu de adresă plat. Într-o mare măsură, adresarea logică a blocurilor facilitează flexibilitatea rețelelor de stocare, permițând integrarea mai ușoară a mai multor tipuri diferite de unități de disc într-un mediu mare de stocare eterogen.

Cu adresarea blocului logic, controlerul unității de disc menține maparea completă a locației tuturor pistelor, sectoarelor și blocurilor din unitatea de disc. Nu există nicio modalitate ca o entitate externă, cum ar fi un sistem de operare sau un controler de subsistem, să știe în ce sector sunt plasate datele sale de către unitatea de disc. La prima vedere, acest lucru poate părea riscant, lăsând un cip mic dintr-o unitate de disc să fie responsabil pentru o funcție atât de importantă. Dar, de fapt, crește fiabilitatea, permițând unității de disc să remapeze sectoarele care au eșuat sau care ar putea fi direcționate în acea direcție.

Având în vedere densitatea ariei și natura microscopică a înregistrării pe disc, vor exista întotdeauna sectoare defectuoase pe orice unitate de disc fabricată. Producătorii de discuri compensează acest lucru rezervând sectoare de rezervă pentru remaparea altor sectoare care nu funcționează corect. Deoarece producătorii anticipează necesitatea sectoarelor de rezervă, capacitatea fizică a unei unități de disc depășește întotdeauna capacitatea logică, utilizabilă. Rezervarea sectoarelor de rezervă pentru remaparea sectoarelor defecte este un produs secundar important, care crește fiabilitatea tehnologiei LBA. Unitățile de disc pot fi fabricate cu sectoare de rezervă amplasate pe toată suprafața platoului, care minimizează performanțele rezultate ale căutării de sectoare remapate.

Geometria unităților de disc și înregistrarea pe biți zonali

Nu există nicio modalitate de a scăpa de geometria radială atunci când lucrați cu unități de disc. Unul dintre aspectele mai interesante ale acestei geometrii radiale este că cantitatea de material de înregistrare dintr-o pistă crește pe măsură ce vă îndepărtați de centrul platoului de disc. Pistele unității de disc pot fi considerate ca inele media având o circumferință care este determinată de expresia matematică 2pr, unde r este raza pistei. Cantitatea de material de înregistrare dintr-o piesă este determinată de lungimea radială. Aceasta înseamnă că pistele exterioare pot conține mai multe date decât piesele interioare. De fapt, pot deține mai multe date decât în ​​interiorul pieselor.

Pentru a profita de această geometrie, proiectanții de unități de disc au dezvoltat înregistrări pe biți zonali, care plasează mai multe sectoare în interiorul pistelor pe măsură ce raza crește. Ideea generală este de a segmenta unitatea în zone „densitate de sector / cale”, unde pistele din acea zonă au toate același număr de sectoare. Zona cea mai exterioară, zona 0, are cele mai multe sectoare pe pistă, în timp ce zona cea mai interioară are cele mai puține.

Adresarea logică a blocurilor facilitează utilizarea înregistrării pe biți zonali, permițând producătorilor de unități de disc să stabilească ce zone doresc, fără să se îngrijoreze de impactul asupra logicii și operațiunilor controlerului gazdă / subsistem. Deoarece platourile nu sunt schimbate niciodată între unitățile de disc, nu este nevoie să vă faceți griji cu privire la configurațiile standardizate ale zonei.

Specificațiile unității de disc

Specificațiile unității de disc pot fi confuze și dificil de interpretat. Această secțiune evidențiază unele dintre cele mai importante specificații utilizate cu unitățile de disc în aplicațiile de rețea de stocare, inclusiv următoarele:

* Timpul mediu între eșecuri
* Viteza de rotație și latența
* Timp mediu de căutare
* Rata de transfer media
* Rata de transfer susținută

Timpul mediu dintre eșecuri

Timpul mediu dintre eșec (MTBF) indică fiabilitatea așteptată a unităților de disc. Specificațiile MTBF sunt derivate folosind metode statistice bine definite și testele rulează pe un număr mare de unități de disc pe o perioadă relativ scurtă de timp. Rezultatele sunt extrapolate și sunt exprimate ca un număr foarte mare de ore, de obicei cuprinse între 500.000 și 1.25 milioane de ore. Aceste cifre sunt de neimaginat de mari pentru unitățile de disc individuale - 1,25 milioane de ore înseamnă aproximativ 135 de ani.

Specificațiile MTBF ajută la crearea așteptărilor cu privire la frecvența cu care vor apărea defecțiunile unității de disc atunci când există multe unități într-un mediu. Folosind specificația MTBF de 1,25 milioane de ore (135 de ani), dacă aveți 135 de unități de disc, vă puteți aștepta să întâmpinați o defecțiune a unității o dată pe an. Într-un mediu de rețea de stocare cu un număr mare de unități de disc - de exemplu, peste 1000 de unități este ușor de văzut că unitățile de rezervă ar trebui să fie disponibile, deoarece aproape sigur vor exista erori ale unității care trebuie gestionate. Acest lucru subliniază, de asemenea, importanța utilizării tehnicilor de redundanță a dispozitivelor de disc, cum ar fi oglindirea sau RAID.

Viteză și latență

Unul dintre cele mai comune moduri de a descrie capacitățile oricărei unități de disc este de a indica viteza de rotație în rpm. Cu cât o unitate de disc se rotește mai repede, cu atât mai repede pot fi scrise și citite date de pe suportul discului. Diferențele de performanță pot fi enorme. Toate celelalte lucruri fiind egale, o unitate de disc de 15.000 rpm poate face mai mult de două ori cantitatea de lucru decât o unitate de disc de 7200 rpm. Dacă 50 sau mai multe unități de disc sunt utilizate de un sistem de procesare a tranzacțiilor, este ușor de văzut de ce cineva ar dori să folosească unități de viteză mai mare.

Legată de viteza de rotație este o specificație numită latență de rotație. După ce capetele unității sunt situate deasupra pistei corespunzătoare într-un platou al unității de disc, acestea trebuie să aștepte ca sectorul corespunzător să treacă dedesubt înainte de a putea efectua transferul de date. Timpul petrecut în așteptarea sectorului potrivit se numește latență de rotație și este direct legat de viteza de rotație a unității de disc.

În esență, latența de rotație este dată ca timpul mediu de așteptare pentru orice operație aleatorie I / O și se calculează ca timpul necesar unui platou pentru a finaliza o jumătate de revoluție.

Latențele de rotație sunt cuprinse între 2 și 6 milisecunde. S-ar putea să nu pară prea mult timp. Dar este foarte lent în comparație cu viteza procesorului și a dispozitivului de memorie. Aplicațiile care tind să sufere de blocaje I / O, cum ar fi procesarea tranzacțiilor, stocarea datelor și transmiterea multimedia necesită unități de disc cu viteze de rotație ridicate și buffere considerabile.

Timp mediu de căutare

Alături de viteza de rotație, timpul de căutare este cea mai importantă specificație de performanță pentru o unitate de disc. Timpul de căutare măsoară timpul necesar actuatorului pentru a repoziționa capetele de citire / scriere de la o pistă la alta peste un platou. Timpii de căutare medii reprezintă o medie de performanță pentru multe operații de I / O și sunt relativ similare cu latența de rotație în intervalul de 4 până la 8 milisecunde.

Procesarea tranzacțiilor și alte aplicații de baze de date care efectuează un număr mare de operații aleatorii l / O în succesiune rapidă necesită unități de disc cu timpi de căutare minimi. Deși este posibil să se împartă volumul de lucru pe mai multe unități, performanța aplicației de tranzacție depinde în mod semnificativ de capacitatea unei unități de disc individuale de a procesa rapid o operație de I / O. Acest lucru se traduce printr-o combinație de timpi de căutare mici și viteze de rotație ridicate.

Rata de transfer media

Rata de transfer media a unei unități de disc măsoară performanța operațiilor de citire / scriere pe biți pe platourile unității. Spre deosebire de majoritatea specificațiilor de stocare, care sunt listate în termeni de octeți, rata de transfer media este dată în termeni de biți. Rata de transfer media măsoară performanța de citire / scriere pe o singură pistă, care depinde de lungimea radială la care este poziționată pista. Cu alte cuvinte, piesele din zona 0 au cele mai rapide rate de transfer media pe unitatea de disc. Din acest motiv, specificațiile ratei de transfer media sunt uneori date folosind intervale.

Rata de transfer susținută

Majoritatea operațiilor de intrare / ieșire pe o unitate de disc funcționează pe mai multe piste și cilindri, ceea ce implică posibilitatea de a schimba locația capetelor de citire / scriere. Specificația ratei de transfer susținute ia în considerare întârzierile fizice ale timpului de căutare și a latenței de rotație și este mult mai aproape de măsurarea performanței reale a datelor utilizatorului decât rata de transfer a mediului.

Acestea fiind spuse, ratele de transfer susținute indică condiții optime dificil de abordat cu aplicațiile reale. Există și alte variabile importante, cum ar fi dimensiunea obiectului mediu de date și nivelul de fragmentare din sistemul de fișiere. Cu toate acestea, rata de transfer susținută este un indiciu destul de bun al performanțelor generale ale unității.

Biletul nr. 16

\*\*Pentru evaluarea 2(doi)la disciplina: Arhitectura calculatoarelor

**Grupele AI 191;SI191;TI1912,3,4,5;FI191 Facultatea C.I.M.**

1. Explicati principiul de sintezare a imaginilor color (RGB).

Modelul cromatic red -green -blue, abreviat RGB, este un model aditiv de culoare, în care culorile albastru, roșu și verde sunt amestecate în diferite moduri pentru a produce o gamă largă de [culori](https://ro.wikipedia.org/wiki/Culoare).

Principalul scop al paletei de culori RGB este de a reprezenta imagini pe sistemele electronice, cum ar fi [televizoarele](https://ro.wikipedia.org/wiki/Televizor) și [calculatoarele](https://ro.wikipedia.org/wiki/Calculator), deși a fost folosit și în fotografie. Înainte de era electronică, modelul RGB era mai puțin cunoscut, în principal el fiind folosit în biologie, la percepția umană a culorilor.

RGB este un model cromatic *dependent de dispozitive*: două monitoare diferite pot reprezenta o anumită valoare RGB diferit, deoarece răspunsul [elementului chimic](https://ro.wikipedia.org/wiki/Element_chimic) ce provoacă culoarea diferă de la producător la producător sau chiar la același dispozitiv, odată cu trecerea timpului. Astfel o valoare RGB nu definește aceeași culoare pe toate dispozitivele fără un fel de gestiune a culorilor.

[Dispozitivele de intrare](https://ro.wikipedia.org/w/index.php?title=Dispozitiv_de_intrare&action=edit&redlink=1) tipice RGB sunt [camerele de filmat](https://ro.wikipedia.org/wiki/Camer%C4%83_video), [scanerele](https://ro.wikipedia.org/wiki/Scaner) și [aparatele de fotografiat](https://ro.wikipedia.org/wiki/Aparat_de_fotografiat), iar cele de ieșire sunt [televizoarele](https://ro.wikipedia.org/wiki/TV) ([CRT](https://ro.wikipedia.org/w/index.php?title=CRT&action=edit&redlink=1), [LCD](https://ro.wikipedia.org/wiki/LCD), cu [plasmă](https://ro.wikipedia.org/w/index.php?title=Ecran_cu_plasm%C4%83&action=edit&redlink=1), etc.), [monitoarele](https://ro.wikipedia.org/wiki/Monitor), ecranele [telefoanelor mobile](https://ro.wikipedia.org/wiki/Telefon_mobil), [proiectoarele video](https://ro.wikipedia.org/w/index.php?title=Proiector&action=edit&redlink=1), afișoarele cu [LED](https://ro.wikipedia.org/wiki/LED)-uri multicolore.

1. Instrucţiunea **JCXZ et** realizează saltul la **et** dacă:
   1. flagul Zero este 1
   2. flagul Zero este 0
   3. flagul Carry este 0
   4. registrul CX este 0
2. Principii de sincronizare a datelor in procesul transmiterii de date.

**Biletul 16**

1. Principiul de bază din spatele afișajelor colorate este că combinarea celor 3 culori de bază - roșu, albastru și verde, poate produce fiecare culoare. Alegând diferite rapoarte ale acestor trei culori putem produce culori diferite - milioane dintre ele de fapt. De asemenea, avem fosfor de bază, care poate produce aceste culori de bază. Deci, ar trebui să existe o tehnologie pentru a le combina în diferite combinații.

Există două tehnici populare pentru producerea afișajelor color cu un CRT:

Metoda de penetrare a fasciculului

Acest CRT este similar cu CRT simplu, dar folosește fosfor multicolor din numărul de straturi. Fiecare strat de fosfor este responsabil pentru o culoare. Toate celelalte aranjamente sunt similare cu CRT simplu. Poate produce maximum 4 până la 5 culori

Organizarea este ceva de genul acesta - Fosforul roșu, verde și albastru este acoperit în straturi - unul în spatele celuilalt. Dacă un fascicul de viteză redusă lovește CRT, se activează doar fosforul de culoare roșie, un fascicul ușor accelerat ar activa atât roșu, cât și verde (deoarece poate pătrunde mai adânc) și unul mult mai activ ar adăuga și componenta albastră.

Dar problema de bază este o tehnologie fiabilă pentru a accelera fasciculul electronic la niveluri precise pentru a obține culorile exacte - este mai ușor de spus decât de făcut. Cu toate acestea, o gamă limitată de culori poate fi produsă convenabil folosind conceptul.

Metoda mască de umbră :

Acest lucru funcționează, din nou, pe principiul combinării culorilor de bază - roșu, verde și albastru - în proporții adecvate pentru a obține o combinație de culori, dar principiul său este mult mai sofisticat și mai stabil.

Masca de umbră CRT, în loc să folosească o armă electronică, folosește 3 arme diferite plasate una lângă cealaltă pentru a forma un triunghi sau o „Deltă”. Fiecare punct de pixel de pe ecran este alcătuit din 3 tipuri de fosfor pentru a produce culori roșii, albastre și verzi. Chiar înainte de ecranul cu fosfor este un ecran metalic, numit „mască de umbră”.

Această placă are găuri plasate strategic, astfel încât atunci când fasciculele de la cele trei tunuri electronice sunt focalizate pe un anumit pixel, acestea se concentrează pe un anumit pixel care produce o culoare, adică dacă din motive de comoditate putem numi fasciculele electronice ca roșu, albastru și verde grinzi (deși în practică culorile sunt produse de fosfor și până când grinzile nu ating punctele de fosfor, acestea nu produc culori), găurile metalice focalizează fasciculul roșu pe culoarea roșie producând fosfor, fasciculul albastru pe cel albastru producând una, etc. Când este focalizat pe un alt pixel, fasciculul roșu se concentrează din nou pe fosforul roșu și așa mai departe.

Acum, spre deosebire de CRT-urile de penetrare a fasciculului în care se monitoriza accelerația fasciculului de electroni, acum manipulăm intensitatea celor 3 fascicule simultan. Dacă fasciculul roșu devine mai intens, obținem mai multă culoare roșie în combinația finală etc. Deoarece reglarea fină a intensităților fasciculului este relativ simplă, putem obține o combinație mult mai mare de culori decât carcasa de penetrare a fasciculului. De fapt, se poate avea o matrice de combinații pentru a produce o mare varietate de culori.

Masca de umbră CRT, deși este mai bună decât performanța CRT de penetrare a fasciculului, nu este lipsită de dezavantaje. Deoarece trei fascicule urmează să fie focalizate, rolul „măștii de umbră” devine critic. Dacă focalizarea nu este realizată corect, rezultatele tind să fie slabe. De asemenea, deoarece în loc de un punct de pixel într-un CRT monocrom acum fiecare pixel este alcătuit din 3 puncte (pentru 3 culori), rezoluția CRT (numărul de pixeli) pentru o anumită dimensiune a ecranului se reduce.

O altă problemă este că, deoarece masca de umbră blochează o porțiune a grinzilor (în timp ce le focalizează prin găuri) intensitățile lor se reduc, reducând astfel luminozitatea generală a imaginii. Pentru a depăși acest efect, grinzile vor trebui să fie produse la intensități foarte mari pentru a începe. De asemenea, deoarece cele 3 puncte de culoare, deși sunt apropiate unele de altele, nu sunt încă în același punct, imaginile tind să arate ca 3 imagini colorate amplasate în apropiere, mai degrabă decât o singură imagine. Desigur, acest efect poate fi redus prin plasarea punctelor cât mai aproape unele de altele.

1. Instrucţiunea **JCXZ et** realizează saltul la **et** dacă:
   1. registrul CX este 0
2. Principii de sincronizare a datelor in procesul transmiterii de date.

Transferurile *izocrone* (*isos* – egal, *chronos* – timp) se utilizează pentru datele care trebuie furnizate cu o anumită rată de transfer constantă şi a căror sincronizare trebuie garantată. Izocron are semnificaţia “cu durată egală” sau “care apare la intervale regulate”. Datele izocrone sunt generate în timp real şi trebuie furnizate cu rata cu care sunt recepţionate pentru a păstra sincronizarea lor. Pe lângă rata de transfer impusă, pentru datele izocrone trebuie respectată şi întârzierea maximă cu care acestea sunt furnizate. Furnizarea la timp a datelor izocrone este asigurată cu preţul unor pierderi potenţiale în şirul de date. Cu alte cuvinte, erorile de transmisie nu sunt corectate prin mecanisme hardware, de exemplu, prin retransmiterea lor. În concluzie, transferurile izocrone se caracterizează prin furnizarea la timp a datelor şi prin lipsa retransmiterii lor în cazul unor erori, deoarece datele întârziate nu mai sunt utile. Spre deosebire de transferurile izocrone, transferurile asincrone se caracterizează prin faptul că fiabilitatea transmiterii datelor este mai importantă decât asigurarea sincronizării. Pentru aceasta se utilizează retransmiterea datelor în cazul unor erori, chiar dacă apar întârzieri din această cauză.

Un exemplu tipic de date izocrone este reprezentat de imaginile video. Dacă rata de transfer a acestor şiruri de date nu este respectată, va avea loc pierderea unor date datorită depăşirii capacităţii bufferelor. Chiar dacă datele sunt furnizate de magistrala USB cu rata adecvată, întârzierile introduse de programe pot afecta negativ aplicaţiile care utilizează aceste date, cum sunt cele pentru videoconferinţe.

Şirurilor de date izocrone li se alocă o porţiune dedicată a lăţimii de bandă a magistralei USB. De asemenea, această magistrală este proiectată pentru o întârziere minimă a transferurilor de date izocrone.

**Biletul 17**

1. **SIMD** (Single Instruction Multiple Data)

Această categorie de arhitecturi cuprinde sistemele de calcul compuse din mai multe unităţi de execuţie identice aflate sub comanda unei singure unităţi de control. Unitatea de control transmite acelaşi flux de instrucţiuni, simultan, tuturor unităţilor de execuţie. Toate unităţile de execuţie execută simultan aceeaşi instrucţiune asupra datelor din memoria proprie (există sisteme ce au şi o memorie partajată pentru comunicaţii). Unitatea de control trebuie să permită tuturor elementelor de procesare să-şi termine instrucţiunea curentă înainte de iniţierea unei noi instrucţiuni, astfel că execuţia instrucţiunilor trebuie sincronizată între toate unităţile de execuţie. Ca şi ordin de mărime numărul procesoarelor implicate într-o structură SIMD este de câteva mii.

Aplicabilitate: calculatoarele SIMD sunt folosite în cazul aplicaţiilor paralele ce necesită un control fin asupra datelor. Exemplu: reţele neuronale.

1. Instrucţiunea **sbb** *dest, sursă* realizează operaţia:
   1. dest<--dest- sursă-carry
2. Principala caracteristică definitorie a IA-32 este disponibilitatea registrelor procesorului de 32 de biți de uz general (de exemplu, EAX și EBX), operații aritmetice și logice întregi pe 32 de biți, compensări pe 32 de biți într-un segment în modul protejat și traducerea adreselor segmentate în adrese liniare pe 32 de biți. Designerii au profitat de ocazie pentru a face și alte îmbunătățiri. Unele dintre cele mai semnificative schimbări sunt descrise mai jos.

Capacitate întregă pe 32 de biți

Toate registrele cu scop general (GPR) sunt extinse de la 16 biți la 32 de biți și toate operațiunile aritmetice și logice, operațiile de memorie-la-înregistrare și înregistrare-la-memorie etc., pot funcționa direct pe numere întregi de 32 de biți. Push-urile și pop-urile din stivă implicit la pași de 4 octeți, iar pointerii nesegmentați au o lățime de 4 octeți.

Moduri de adresare mai generale

Orice GPR poate fi utilizat ca registru de bază și orice alt GPR decât ESP poate fi folosit ca registru index, într-o referință de memorie. Valoarea registrului index poate fi înmulțită cu 1, 2, 4 sau 8 înainte de a fi adăugată la valoarea registrului de bază și deplasarea.

Registre de segmente suplimentare

Sunt furnizate două registre de segmente suplimentare, FS și GS.

Spațiu de adresă virtual mai mare

Arhitectura IA-32 definește un format de adresă segmentat pe 48 de biți, cu un număr de segment pe 16 biți și un offset pe 32 de biți în cadrul segmentului. Adresele segmentate sunt mapate la adrese liniare pe 32 de biți.

Cerere de paginare

Adresele liniare pe 32 de biți sunt mai degrabă adrese virtuale decât adrese fizice; sunt traduse în adrese fizice printr-un tabel de pagini. În procesoarele 80386, 80486 și originale Pentium, adresa fizică era de 32 de biți; în procesoarele Pentium Pro și ulterior, extensia de adresă fizică a permis adresele fizice de 36 de biți, deși dimensiunea liniară a adresei era încă de 32 de biți.

**Biletul 18**

1. O stivă este un tip de date abstract care servește ca o colecție de elemente, cu două operații principale principale:

* Push, care adaugă un element colecției și
* Pop, care elimină cel mai recent element adăugat care nu a fost încă eliminat.

Ordinea în care elementele se desprind dintr-o stivă dă naștere denumirii sale alternative, LIFO (ultima intrare, prima ieșire). În plus, o operațiune peek poate da acces la partea de sus fără a modifica stiva. Denumirea de „stivă” pentru acest tip de structură vine de la analogia cu un set de articole fizice stivuite unul peste altul. Această structură face mai ușor să scoateți un articol de pe partea de sus a teancului, în timp ce pentru a ajunge la un articol mai adânc în teanc poate fi necesar să scoateți mai întâi mai multe alte obiecte.

Considerate ca o structură de date liniară sau mai abstract o colecție secvențială, operațiile push și pop au loc doar la un capăt al structurii, denumit partea de sus a stivei. Această structură de date face posibilă implementarea unei stive ca o listă legată individual și un indicator către elementul de sus. O stivă poate fi implementată pentru a avea o capacitate mărginită. Dacă stiva este plină și nu conține suficient spațiu pentru a accepta o entitate care trebuie împinsă, stiva este atunci considerată a fi într-o stare de revărsare. Operația pop elimină un articol din partea de sus a stivei.

1. ***Instructiunea CALL* (apel de procedura)**

Poate apărea sub una din formele:

*CALL nume\_proc*

*CALL NEAR PTR nume\_proc*

*CALL FAR PTR nume\_proc*

Tipul apelului poate fi dedus din tipul procedurii (primul caz) sau specificat explicit prin NEAR şi FAR. Tipul apelului trebuie să coincidă cu tipul procedurii şi cu tipul instructiunii RETURN din interiorul procedurii.

1. Un computer analog este un tip de computer care folosește aspectele continuu schimbabile ale fenomenelor fizice, cum ar fi mărimile electrice, mecanice sau hidraulice pentru a modela problema rezolvată. În schimb, calculatoarele digitale reprezintă cantități variate simbolic și prin valori discrete atât de timp cât și de amplitudine.

Calculatoarele analogice pot avea o gamă foarte largă de complexitate. Regulile de diapozitive și nomogramele sunt cele mai simple, în timp ce computerele navale de control al focurilor de armă și calculatoarele hibride digitale / analogice mari erau printre cele mai complicate. Sistemele pentru controlul proceselor și releele de protecție au folosit calcule analogice pentru a efectua funcții de control și protecție.

Majoritatea computerelor analogice electronice actuale funcționează prin manipularea diferențelor de potențial (tensiuni). Componenta lor de bază este un amplificator operațional, un dispozitiv al cărui curent de ieșire este proporțional cu diferența de potențial de intrare. Prin determinarea acestui curent de ieșire să curgă prin componentele adecvate, se obțin diferențe potențiale suplimentare și se poate efectua asupra acestora o mare varietate de operații matematice, inclusiv inversare, însumare, diferențiere și integrare. Un computer analogic tipic electronic constă în numeroase tipuri de amplificatoare, care pot fi conectate astfel încât să construiască o expresie matematică, uneori de mare complexitate și cu o multitudine de variabile.

Calculatoarele analogice sunt deosebit de potrivite pentru simularea sistemelor dinamice; astfel de simulări pot fi efectuate în timp real sau la viteze foarte accelerate, permițând astfel experimentarea prin repetări cu variabile modificate. Acestea au fost utilizate pe scară largă în simulări de avioane, centrale nucleare și procese chimice industriale. Alte utilizări majore includ analiza rețelelor hidraulice (de exemplu, fluxul de lichide printr-un sistem de canalizare) și a rețelelor electronice (de exemplu, performanța circuitelor la distanță).

**Biletul 19**

1. Registrul notat FLAGS cuprinde flagurile (biţi indicatori) procesorului, sau bistabililor de condiţie, iar registrul IP (instruction pointer) este registrul de instrucțiuni.
2. Instrucţiunea **add dest**, **sursă** realizează operaţia:

dest<--dest+ sursă

1. Memoria cache este o memorie specială de mare viteză. Este folosit pentru a accelera și sincroniza cu CPU de mare viteză. Memoria cache este mai costisitoare decât memoria principală sau memoria discului, dar mai economică decât registrele CPU. Memoria cache este un tip de memorie extrem de rapid care acționează ca un buffer între RAM și CPU. Păstrează datele și instrucțiunile solicitate frecvent, astfel încât acestea să fie disponibile imediat procesorului atunci când este necesar.

Memoria cache este utilizată pentru a reduce timpul mediu de accesare a datelor din memoria principală. Memoria cache este o memorie mai mică și mai rapidă, care stochează copii ale datelor din locațiile de memorie principale utilizate frecvent. Există diferite cache-uri independente diferite într-un procesor, care stochează instrucțiuni și date.

Niveluri de memorie:

* Nivelul 1 sau Înregistrare -

Este un tip de memorie în care sunt stocate și acceptate date care sunt stocate imediat în CPU. Cel mai frecvent utilizat registru este acumulatorul, contorul de programe, registrul de adrese etc.

* Nivelul 2 sau memoria cache -

Este cea mai rapidă memorie care are un timp de acces mai rapid în care datele sunt stocate temporar pentru un acces mai rapid.

* Nivelul 3 sau memorie principală -

Este memoria pe care computerul funcționează în prezent. Are dimensiuni reduse și odată ce alimentarea este oprită, datele nu mai rămân în această memorie.

* Nivelul 4 sau memoria secundară -

Este memorie externă care nu este la fel de rapidă ca memoria principală, dar datele rămân permanent în această memorie.

Performanță cache:

Când procesorul trebuie să citească sau să scrie o locație în memoria principală, verifică mai întâi o intrare corespunzătoare în cache.

* Dacă procesorul constată că locația memoriei se află în cache, a avut loc o lovire cache și datele sunt citite din cache
* Dacă procesorul nu găsește locația de memorie în cache, a apărut o pierdere a memoriei cache. Pentru o pierdere a cache-ului, cache-ul alocă o nouă intrare și copiază datele din memoria principală, apoi cererea este îndeplinită din conținutul cache-ului.

Cartografierea memoriei cache:

Există trei tipuri diferite de mapare utilizate în scopul memorării cache care sunt după cum urmează: Mapare directă, mapare asociativă și mapare set-asociativă.

* Cartografiere directă -

Cea mai simplă tehnică, cunoscută sub numele de mapare directă, mapează fiecare bloc de memorie principală într-o singură linie cache posibilă. sau

În maparea directă, atribuiți fiecare bloc de memorie unei linii specifice din cache. Dacă o linie este preluată anterior de un bloc de memorie atunci când trebuie încărcat un bloc nou, blocul vechi este în coșul de gunoi. Un spațiu de adrese este împărțit în două părți câmp index și un câmp etichetă. Cache-ul este utilizat pentru a stoca câmpul de etichetă, în timp ce restul este stocat în memoria principală. Performanța cartografierii directe este direct proporțională cu raportul Hit.

* Cartografiere asociativă -

În acest tip de mapare, memoria asociativă este utilizată pentru a stoca conținutul și adresele cuvântului de memorie. Orice bloc poate intra în orice linie a cache-ului. Aceasta înseamnă că cuvântul biți de identificare este utilizat pentru a identifica ce cuvânt din bloc este necesar, dar eticheta devine toți biții rămași. Aceasta permite plasarea oricărui cuvânt în orice loc din memoria cache. Este considerat a fi cel mai rapid și cel mai flexibil formular de cartografiere.

* Cartografiere asociativă -

Această formă de cartografiere este o formă îmbunătățită de cartografiere directă în care sunt eliminate dezavantajele cartografierii directe. Setați asociativ abordează problema unei posibile bătăi în metoda de mapare directă. Face acest lucru spunând că, în loc să avem exact o linie pe care un bloc să o mapeze în cache, vom grupa câteva linii împreună creând un set. Apoi, un bloc din memorie poate mapa pe oricare dintre liniile unui anumit set..Set-maparea asociativă permite ca fiecare cuvânt care este prezent în cache să aibă două sau mai multe cuvinte în memoria principală pentru aceeași adresă de index. Setarea cartografierii cache asociative combină cele mai bune tehnici de cartografiere cache directă și asociativă.

În acest caz, cache-ul constă dintr-un număr de seturi, fiecare dintre ele constând dintr-un număr de linii.

Tipuri de cache :

* Cache principal

O memorie cache principală este întotdeauna localizată pe cipul procesorului. Această memorie cache este mică și timpul său de acces este comparabil cu cel al registrelor procesorului.

* Cache secundar

Memoria cache secundară este plasată între memoria cache principală și restul memoriei. Este denumit cache de nivel 2 (L2). Adesea, memoria cache de nivel 2 este găzduită și pe cipul procesorului.

Localitatea de referință -

Deoarece dimensiunea memoriei cache este mai mică în comparație cu memoria principală. Deci, pentru a verifica care parte a memoriei principale ar trebui să i se acorde prioritate și să se încarce în cache se decide pe baza localității de referință.

Tipuri de localitate de referință:

* Localitate spațială de referință

Acest lucru spune că există șansa ca elementul să fie prezent în imediata apropiere a punctului de referință și data viitoare dacă este căutat din nou, apoi mai aproape de punctul de referință.

* Localitate temporală de referință

În acest algoritm cel mai puțin folosit recent va fi folosit. Ori de câte ori apare o eroare de pagină într-un cuvânt, nu numai că va încărca cuvântul în memoria principală, dar va fi încărcată eroarea de pagină completă, deoarece localitatea spațială a regulii de referință spune că, dacă vă referiți la orice cuvânt, următorul cuvânt va fi menționat în registrul său, de aceea încărcăm completează tabelul de pagini, astfel încât blocul complet să fie încărcat.

**Biletul 20**

1. Conceptul de memorie virtuală se bazează pe ideea separării memoriei logice a utilizatorului de cea fizică și extinderea memoriei logice prin stocare imaginii sale pe hard disc (fișierul pagefile.sys). Fiecare program la execuție, nu este obligatoriu să fie stocat întreg în memoria RAM, ci doar o secvență de cod și date, executate la un moment dat. Apare, în mod firesc, ideea de a extinde spațiul de memorie logică, care poate fi realizat cu mult mai mare decât memoria fizică. Acesta este principiul de bază al organizării memoriei virtuale.

Legat de conceptul memoriei virtuale apare mecanismul de gestionare a memoriei. El constă în translatarea adreselor virtuale (folosite de programator) în adrese fizice identificabile în structura fizică, concretă, a microcalculatorului.

1. 00001110
2. Unitatea de control este partea unității centrale de procesare (CPU) a computerului, care direcționează funcționarea procesorului. A fost inclus ca parte a Arhitecturii Von Neumann de John von Neumann. Este responsabilitatea Unității de control să informeze memoria computerului, unitatea aritmetică / logică și dispozitivele de intrare și ieșire cum să răspundă la instrucțiunile trimise procesorului. Acesta preia instrucțiuni interne ale programelor din memoria principală în registrul de instrucțiuni al procesorului și, pe baza conținutului acestui registru, unitatea de control generează un semnal de control care supraveghează executarea acestor instrucțiuni.

O unitate de control funcționează primind informații de intrare la care se convertește în semnale de control, care sunt apoi trimise procesorului central. Procesorul computerului spune apoi hardware-ului atașat ce operațiuni trebuie efectuate. Funcțiile pe care le efectuează o unitate de control depind de tipul de CPU, deoarece arhitectura CPU variază de la producător la producător. Exemple de dispozitive care necesită un CU sunt:

* Unități de procesare a controlului (CPU)
* Unități de procesare grafică (GPU-uri)

Funcțiile unității de control

* Coordonează secvența mișcărilor de date în, din și între multele subunități ale procesorului.
* Interpretează instrucțiunile.
* Controlează fluxul de date în interiorul procesorului.
* Primește instrucțiuni sau comenzi externe la care se convertește în succesiune de semnale de control.
* Controlează multe unități de execuție (adică ALU, tampoane de date și registre) conținute într-un procesor.
* De asemenea, gestionează mai multe sarcini, cum ar fi preluarea, decodarea, gestionarea execuției și stocarea rezultatelor.

Tipuri de unități de control

Există două tipuri de unități de control: unitatea de control cablată și unitatea de control microprogramabilă.

Unitate de control prin cablu -

În unitatea de control Hardwired, semnalele de control care sunt importante pentru controlul executării instrucțiunilor sunt generate de circuite logice hardware special concepute, în care nu putem modifica metoda de generare a semnalului fără schimbarea fizică a structurii circuitului. Codul de funcționare al unei instrucțiuni conține datele de bază pentru generarea semnalului de control. În decodorul de instrucțiuni, codul de funcționare este decodat. Decodorul de instrucțiuni constituie un set de multe decodificatoare care decodează diferite câmpuri ale codului de instrucțiuni.

Ca rezultat, puține linii de ieșire care ies din decodorul de instrucțiuni obțin valori de semnal active. Aceste linii de ieșire sunt conectate la intrările matricei care generează semnale de control pentru unitățile executive ale computerului. Această matrice implementează combinații logice ale semnalelor decodificate din instrucțiunea opcode cu ieșirile din matricea care generează semnale reprezentând stări consecutive ale unității de control și cu semnale care vin din exteriorul procesorului, de ex. semnale de întrerupere. Matricile sunt construite într-un mod similar cu matricile logice programabile.

Unitate de control microprogramabilă -

Diferența fundamentală dintre aceste structuri de unitate și structura unității de control cablate este existența depozitului de control care este utilizat pentru stocarea cuvintelor care conțin semnale de control codificate obligatorii pentru executarea instrucțiunilor.

În unitățile de control microprogramate, cuvintele de instrucțiuni ulterioare sunt preluate în registrul de instrucțiuni într-un mod normal. Cu toate acestea, codul de funcționare al fiecărei instrucțiuni nu este direct decodat pentru a permite generarea imediată a semnalului de control, ci cuprinde adresa inițială a unui microprogram conținut în depozitul de control.

**Biletul 21**

1. O întrerupere programată este un răspuns al procesorului la un eveniment care necesită atenție din partea software-ului. O condiție de întrerupere alertează procesorul și servește ca o cerere pentru procesor de a întrerupe codul de executare curent când este permis, astfel încât evenimentul să poată fi procesat în timp util. Dacă cererea este acceptată, procesorul răspunde suspendându-și activitățile curente, salvându-și starea și executând o funcție numită un handler de întrerupere (sau o rutină de servicii de întrerupere, ISR) pentru a face față evenimentului. Această întrerupere este temporară și, cu excepția cazului în care întreruperea indică o eroare fatală, procesorul reia activitățile normale după terminarea handlerului de întreruperi.

În concluzie, întreruperea este un semnal emis de hardware sau software atunci când un proces sau un eveniment are nevoie de atenție imediată. Alertează procesorul cu privire la un proces cu prioritate ridicată care necesită întreruperea procesului de lucru curent.

1. Dacă în registrul AL este stocată valoarea 11001001, după Instrucţiunea **shl al,3** se va obţine:

01001000

1. O unitate logică aritmetică (ALU) este un circuit digital combinațional care efectuează operații aritmetice și în biți pe numere binare întregi. Acest lucru este în contrast cu o unitate cu virgulă mobilă (FPU), care operează pe numere în virgulă mobilă. Este un element fundamental al multor tipuri de circuite de calcul, inclusiv unitatea centrală de procesare (CPU) a computerelor, FPU-urilor și unităților de procesare grafică (GPU-uri).

Intrările într-o ALU sunt datele care trebuie operate, numiți operanzi și un cod care indică operația care trebuie efectuată; ieșirea ALU este rezultatul operației efectuate. În multe modele, ALU are, de asemenea, intrări sau ieșiri de stare, sau ambele, care transmit informații despre o operație anterioară sau respectiv operațiunea curentă, între ALU și registrele de stare externe.

**Biletul 22**

1. Modelul de memorie plană sau modelul de memorie liniară se referă la o paradigmă de adresare a memoriei în care „memoria apare programului ca un singur spațiu de adresare contiguu.” CPU poate adresa direct (și liniar) toate locațiile de memorie disponibile fără a fi nevoie să recurge la orice fel de scheme de segmentare a memoriei sau de paginare.

Gestionarea memoriei și traducerea adreselor pot fi implementate în continuare deasupra unui model de memorie plană pentru a facilita funcționalitatea sistemului de operare, protecția resurselor, multitasking sau pentru a crește capacitatea de memorie dincolo de limitele impuse de spațiul de adrese fizic al procesorului, dar caracteristica cheie a unui model de memorie plană este că întregul spațiu de memorie este liniar, secvențial și contigu.

Într-un controler simplu sau într-o singură aplicație încorporată cu sarcini, în care gestionarea memoriei nu este necesară și nici nu este de dorit, modelul de memorie plat este cel mai potrivit, deoarece oferă cea mai simplă interfață din punctul de vedere al programatorului, cu acces direct la toată memoria locații și complexitate minimă de proiectare.

Într-un sistem informatic de uz general, care necesită multitasking, alocarea resurselor și protecție, sistemul de memorie plat trebuie să fie mărit de o schemă de gestionare a memoriei, care este de obicei implementată printr-o combinație de hardware dedicat (în interiorul sau în afara procesorului) și software construit în sistemul de operare. Modelul de memorie plană (la nivelul adresării fizice) oferă în continuare cea mai mare flexibilitate pentru implementarea acestui tip de gestionare a memoriei.

1. Instrucţiunea **JCXZ et** realizează saltul la **et** dacă:
   1. registrul CX este 0
2. În calcul, o mașină virtuală (VM) este virtualizarea / emularea unui sistem computerizat. Mașinile virtuale se bazează pe arhitecturi de calculatoare și oferă funcționalitatea unui computer fizic. Implementările lor pot implica hardware specializat, software sau o combinație.

Mașinile virtuale diferă și sunt organizate după funcția lor, prezentată aici:

* Mașinile virtuale de sistem (numite și VM-uri de virtualizare completă) oferă un substitut pentru o mașină reală. Acestea oferă funcționalități necesare pentru a executa sisteme de operare întregi. Un hipervizor folosește execuția nativă pentru a partaja și gestiona hardware, permițând medii multiple care sunt izolate unele de altele, dar există pe aceeași mașină fizică. Hipervizorii moderni utilizează virtualizarea asistată de hardware, hardware specific virtualizării, în principal de la procesoarele gazdă.
* Mașinile virtuale de proces sunt concepute pentru a executa programe de calculator într-un mediu independent de platformă.

Unele emulatoare de mașini virtuale, cum ar fi QEMU și emulatoarele de console de jocuri video, sunt concepute pentru a emula (sau „imita practic”) diferite arhitecturi de sistem, permițând astfel executarea aplicațiilor software și a sistemelor de operare scrise pentru un alt procesor sau arhitectură. Virtualizarea la nivel de sistem de operare permite partiționarea resurselor unui computer prin kernel. Termenii nu sunt universal interschimbabili.

**Biletul 23**

1. Un compilator este un program de computer care transformă codul scris într-un limbaj de programare la nivel înalt în codul mașinii. Este un program care traduce codul care poate fi citit de om într-un limbaj pe care un procesor de computer îl înțelege (binar 1 și 0 biți). Calculatorul procesează codul mașinii pentru a efectua sarcinile corespunzătoare.

Un compilator ar trebui să respecte regula de sintaxă a acelui limbaj de programare în care este scris. Cu toate acestea, compilatorul este doar un program și nu poate remedia erorile găsite în acel program.

Un interpretor(interpreter) este un program de calculator, care acoperă fiecare instrucțiune de program la nivel înalt în codul mașinii. Aceasta include codul sursă, codul precompilat și scripturile. Atât compilatorul, cât și interpretorii fac aceeași treabă, care este convertirea limbajului de programare de nivel superior în codul mașinii. Cu toate acestea, un compilator va converti codul în codul mașinii (va crea un exe) înainte de a rula programul. Interpretorii convertesc codul în codul mașinii atunci când programul este rulat.

Rolul compilatorului

* Furnizorii citesc codul sursă, scoate codul executabil
* Traduce software-ul scris într-un limbaj de nivel superior în instrucțiuni pe care computerul le poate înțelege. Convertește textul pe care un programator îl scrie într-un format pe care CPU îl poate înțelege.
* Procesul de compilare este relativ complicat. Petrece mult timp analizând și procesând programul.
* Rezultatul executabil este o formă de cod binar specific mașinii.

Rolul interpretorului:

* Interpretorul convertește codul sursă linie cu linie în timpul RUN Time.
* Interpretorul traduce complet un program scris într-un limbaj la nivel înalt în limbaj la nivel de mașină.
* Interpretorul permite evaluarea și modificarea programului în timpul executării acestuia.
* Relativ mai puțin timp petrecut pentru analiza și procesarea programului
* Executarea programului este relativ lentă în comparație cu compilatorul

1. Instrucţiunea **in al,71h:**
   1. scrie valoarea din registrul **al** la portul 71h
2. RGB (rosu-verde-albastru) este un model de reprezentare a culorilor in cadrul unei imagini. Fiecare culoare utilizata in cadrul acestui model este formata din 3 octeti, care reprezinta la randul lor scara utilizarii fiecarei culori: rosu, verde si albastru. Se poate reprezenta modelul RGB sub forma unui cub, in spatiul tridimensional, in care fiecare axa reprezinta fiecare din cele 3 culori: albastru (axa y, cu coordonatele 0, 0, 1), rosu (axa z, cu coordonatele 1,0,0) si verde (axa y, cu coordonatele 0,1,0). Orginea corespunde culoarea neagra (0, 0, 0). Diagonala acestui cub, de la origine la culoarea alba (1,1,1) corespunde paletei de culori gri. Astfel, fiecarui pixel al imaginii ii corepunde 3 octeti ai valorilor culorilor RGB, care formeaza o culoare din cele 3 culori de baza.

Principiul de bază din spatele afișajelor colorate este că combinarea celor 3 culori de bază - roșu, albastru și verde, poate produce fiecare culoare. Alegând diferite rapoarte ale acestor trei culori putem produce culori diferite - milioane dintre ele de fapt. De asemenea, avem fosfor de bază, care poate produce aceste culori de bază. Deci, ar trebui să existe o tehnologie pentru a le combina în diferite combinații.

Există două tehnici populare pentru producerea afișajelor color cu un CRT:

Metoda de penetrare a fasciculului

Acest CRT este similar cu CRT simplu, dar folosește fosfor multicolor din numărul de straturi. Fiecare strat de fosfor este responsabil pentru o culoare. Toate celelalte aranjamente sunt similare cu CRT simplu. Poate produce maximum 4 până la 5 culori

Organizarea este ceva de genul acesta - Fosforul roșu, verde și albastru este acoperit în straturi - unul în spatele celuilalt. Dacă un fascicul de viteză redusă lovește CRT, se activează doar fosforul de culoare roșie, un fascicul ușor accelerat ar activa atât roșu, cât și verde (deoarece poate pătrunde mai adânc) și unul mult mai activ ar adăuga și componenta albastră.

Dar problema de bază este o tehnologie fiabilă pentru a accelera fasciculul electronic la niveluri precise pentru a obține culorile exacte - este mai ușor de spus decât de făcut. Cu toate acestea, o gamă limitată de culori poate fi produsă convenabil folosind conceptul.

Metoda mască de umbră :

Acest lucru funcționează, din nou, pe principiul combinării culorilor de bază - roșu, verde și albastru - în proporții adecvate pentru a obține o combinație de culori, dar principiul său este mult mai sofisticat și mai stabil.

Masca de umbră CRT, în loc să folosească o armă electronică, folosește 3 arme diferite plasate una lângă cealaltă pentru a forma un triunghi sau o „Deltă”. Fiecare punct de pixel de pe ecran este alcătuit din 3 tipuri de fosfor pentru a produce culori roșii, albastre și verzi. Chiar înainte de ecranul cu fosfor este un ecran metalic, numit „mască de umbră”.

Această placă are găuri plasate strategic, astfel încât atunci când fasciculele de la cele trei tunuri electronice sunt focalizate pe un anumit pixel, acestea se concentrează pe un anumit pixel care produce o culoare, adică dacă din motive de comoditate putem numi fasciculele electronice ca roșu, albastru și verde grinzi (deși în practică culorile sunt produse de fosfor și până când grinzile nu ating punctele de fosfor, acestea nu produc culori), găurile metalice focalizează fasciculul roșu pe culoarea roșie producând fosfor, fasciculul albastru pe cel albastru producând una, etc. Când este focalizat pe un alt pixel, fasciculul roșu se concentrează din nou pe fosforul roșu și așa mai departe.

Acum, spre deosebire de CRT-urile de penetrare a fasciculului în care se monitoriza accelerația fasciculului de electroni, acum manipulăm intensitatea celor 3 fascicule simultan. Dacă fasciculul roșu devine mai intens, obținem mai multă culoare roșie în combinația finală etc. Deoarece reglarea fină a intensităților fasciculului este relativ simplă, putem obține o combinație mult mai mare de culori decât carcasa de penetrare a fasciculului. De fapt, se poate avea o matrice de combinații pentru a produce o mare varietate de culori.

Masca de umbră CRT, deși este mai bună decât performanța CRT de penetrare a fasciculului, nu este lipsită de dezavantaje. Deoarece trei fascicule urmează să fie focalizate, rolul „măștii de umbră” devine critic. Dacă focalizarea nu este realizată corect, rezultatele tind să fie slabe. De asemenea, deoarece în loc de un punct de pixel într-un CRT monocrom acum fiecare pixel este alcătuit din 3 puncte (pentru 3 culori), rezoluția CRT (numărul de pixeli) pentru o anumită dimensiune a ecranului se reduce.

O altă problemă este că, deoarece masca de umbră blochează o porțiune a grinzilor (în timp ce le focalizează prin găuri) intensitățile lor se reduc, reducând astfel luminozitatea generală a imaginii. Pentru a depăși acest efect, grinzile vor trebui să fie produse la intensități foarte mari pentru a începe. De asemenea, deoarece cele 3 puncte de culoare, deși sunt apropiate unele de altele, nu sunt încă în același punct, imaginile tind să arate ca 3 imagini colorate amplasate în apropiere, mai degrabă decât o singură imagine. Desigur, acest efect poate fi redus prin plasarea punctelor cât mai aproape unele de altele.

**Biletul 24**

1. Amestecând roșu, verde și albastru (primarele aditive) în diferite combinații și la diferite niveluri de intensitate, putem simula întreaga gamă de culori din natură. Dacă lumina reflectată conține un amestec de lumină roșie pură, verde și albastră, ochiul percepe albul. Când nu există lumină, ochiul percepe negru.

Combinarea a două elemente primare aditive pure produce un element primar subtractiv. Primarele subtractive ale cianului, magenta și galbenului sunt culorile opuse roșu, verde și albastru.

Televizoarele, telefoanele mobile, tabletele și monitoarele de computer folosesc sistemul de culoare aditivă, deoarece sunt dispozitive emisive. Încep cu întunericul și adaugă lumină roșie, verde și albastră pentru a crea spectrul de culori.

Metodele de redare a culorilor utilizate de aceste dispozitive se bazează direct pe răspunsul nostru la stimulii de lumină roșie, verde și albastră. La fel ca ochiul uman, aceste dispozitive trebuie, de asemenea, să proceseze o cantitate mare de informații despre culoare simultan - pe ecran. În mod logic, aceste dispozitive imită răspunsul ochiului la elementele primare aditive pentru a crea o iluzie colorată.

De exemplu, un monitor de computer amestecă intensități variate de lumină roșie, verde și albastră la fiecare dintre micii săi pixeli. Acești pixeli sunt atât de mici și strânși, încât răspunsul RGB al ochiului este „păcălit” în percepția multor culori diferite, atunci când există doar trei.

CMY și CMYK — Primare subtractive

Imprimantele, pe de altă parte, redă culorile pe hârtie și alte substraturi, deci trebuie să lucreze cu lumina reflectată. Pentru a face acest lucru, ei folosesc elementele primare subtractive opuse de cian, magenta și galben.

În spectrul vizibil, cianul este opus direct roșului; magenta este opusul verde; iar galbenul este opusul albastrului. Când pigmenții cian, magenta și galben sunt așezați pe un substrat alb, reflectorizant, fiecare absoarbe complet - sau scade - omologul său opus din lumina albă. Procesele de imprimare utilizează cerneluri cyan, magenta și galben pentru a controla cantitatea de lumină roșie, verde și albastră reflectată de hârtia albă.

Când două elemente primare subtractive se suprapun, se produce un element primar aditiv.

La imprimarea color subtractivă, se adaugă o a patra culoare, negru (K, care înseamnă cheie) pentru a face imprimarea în patru culori (CMYK). Dacă am folosi doar cian, magenta și galben pentru a face negru, am obține o culoare maronie datorită impurităților din acele culori de cerneală. Cerneala neagră ajută la neutralizarea imaginilor și graficelor și adaugă densitate umbrelor.

1. Complementul față de doi se utilizează pentru:
   1. reprezentarea numerelor întregi negative
2. Controllerul DMA (DMA-direct memory access, acces direct la memorie) este un dispozitiv ce asigură accesul direct la RAM a perifericelor, neutilizând registrii interni ai microprocesorului.

Cererea de acces este formulată pe un terminal specializat denumit, de regulă, "BUSRQ" (cerere de magistrală). Are prioritatea maximă şi răspunsul aşteaptă doar terminarea ciclului maşină curent.

Microprocesorul "îngheaţă" întreaga activitate internă; magistralele sale trec în "impedanţă înaltă" iar controlul transferului de informație este preluat de controllerul DMA. Acesta facilitează transferul informaţiei direct între memorie şi echipamente periferice.

Controllerul de DMA poate fi extern sau intern unui microprocesor standard. El adresează blocuri de date în memorie (succesiv, între limite prestabilite) şi comandă transferul la sau de la un periferic prestabilit.

Activitatea de DMA este luată drept etalon pentru viteza maximă de circulaţie a datelor pe magistrala unui calculator (microprocesorul nu participă şi deci adresarea memoriei şi perifericelor nu este afectată de timpul de calcul al adresei fizice).

Accesul direct la memorie (DMA) este o caracteristică a sistemelor informatice care permite anumitor subsisteme hardware să acceseze memoria principală a sistemului (memorie cu acces aleatoriu) independent de unitatea centrală de procesare (CPU).

Fără DMA, atunci când CPU utilizează intrare / ieșire programată, acesta este de obicei complet ocupat pe întreaga durată a operației de citire sau scriere și, prin urmare, nu este disponibil pentru a efectua alte lucrări. Cu DMA, CPU inițiază mai întâi transferul, apoi face alte operații în timp ce transferul este în desfășurare și, în cele din urmă, primește o întrerupere de la controlerul DMA (DMAC) când operațiunea este terminată. Această caracteristică este utilă în orice moment în care CPU nu poate ține pasul cu rata de transfer de date sau când CPU trebuie să efectueze lucrări în timp ce așteaptă un transfer de date I / O relativ lent. Multe sisteme hardware folosesc DMA, inclusiv controlere pentru unități de disc, plăci grafice, plăci de rețea și plăci de sunet. DMA este, de asemenea, utilizat pentru transferul de date intra-cip în procesoare multi-core. Computerele care au canale DMA pot transfera date către și de la dispozitive cu o suprafață CPU mult mai mică decât computerele fără canale DMA. În mod similar, un element de procesare din interiorul unui procesor multi-core poate transfera date către și din memoria sa locală fără a ocupa timpul procesorului său, permițând calculului și transferului de date să se desfășoare în paralel.

DMA poate fi, de asemenea, utilizat pentru „memorie în memorie” pentru copierea sau mutarea datelor în memorie. DMA poate descărca operațiuni de memorie scumpe, cum ar fi copii mari sau operații de împrăștiere, de la CPU la un motor DMA dedicat. Un exemplu de implementare este I / O Acceleration Technology. DMA prezintă interes pentru arhitecturile de calcul rețea-pe-cip și în memorie.

**Biletul 25**

1. Un compilator este un program de computer care transformă codul scris într-un limbaj de programare la nivel înalt în codul mașinii. Este un program care traduce codul care poate fi citit de om într-un limbaj pe care un procesor de computer îl înțelege (binar 1 și 0 biți). Calculatorul procesează codul mașinii pentru a efectua sarcinile corespunzătoare.

Un compilator ar trebui să respecte regula de sintaxă a acelui limbaj de programare în care este scris. Cu toate acestea, compilatorul este doar un program și nu poate remedia erorile găsite în acel program.

Un interpretor(interpreter) este un program de calculator, care acoperă fiecare instrucțiune de program la nivel înalt în codul mașinii. Aceasta include codul sursă, codul precompilat și scripturile. Atât compilatorul, cât și interpretorii fac aceeași treabă, care este convertirea limbajului de programare de nivel superior în codul mașinii. Cu toate acestea, un compilator va converti codul în codul mașinii (va crea un exe) înainte de a rula programul. Interpretorii convertesc codul în codul mașinii atunci când programul este rulat.

Rolul compilatorului

* Furnizorii citesc codul sursă, scoate codul executabil
* Traduce software-ul scris într-un limbaj de nivel superior în instrucțiuni pe care computerul le poate înțelege. Convertește textul pe care un programator îl scrie într-un format pe care CPU îl poate înțelege.
* Procesul de compilare este relativ complicat. Petrece mult timp analizând și procesând programul.
* Rezultatul executabil este o formă de cod binar specific mașinii.

Rolul interpretorului:

* Interpretorul convertește codul sursă linie cu linie în timpul RUN Time.
* Interpretorul traduce complet un program scris într-un limbaj la nivel înalt în limbaj la nivel de mașină.
* Interpretorul permite evaluarea și modificarea programului în timpul executării acestuia.
* Relativ mai puțin timp petrecut pentru analiza și procesarea programului
* Executarea programului este relativ lentă în comparație cu compilatorul

1. Instrucţiunea **cmp** *dest*, *sursă* realizează operaţia:

compara operanzii - sursă si destinaţie prin scădere

1. La banda de asamblare cu 6 segmente a procesorului intrucțiunile se execută secvențial, dar nu pot fi suprapuneri între ele. Plăcile video afișează la monitor datele prelucrate de CPU.