

SZÁMÍTÓGÉP-ARCHITEKTÚRÁK I

1. ZH

A hardver fogalma, fejlődése és leírása

1. Határozza meg a **hardver fogalmát**!

Lényegében ez a számítógépet alkotó, kézzel fogható eszközök összefoglaló neve, a számítógép elektronikus áramköreit, mechanikus berendezéseit, kábeleit, csatlakozásait és perifériáit nevezzük így.

2. Miért van szükség a **hardver rétegmodelljére**?

Például a processzor tranzisztorokból való felépítését kapcsolási rajzokkal dokumentáljuk. Ez csak olyan tömegű papírmennyiséggel lenne megoldható, ami emberileg áttekinthetetlen adathalmaz. A megoldás csak az lehet, ha a hardvert hierarchikus szintekkel próbáljuk modellezni, amelynél egy alsóbb rétegbeli építőelemekből épülnek fel a következő réteg hardveregységei.

3. Sorolja fel a **hardver rétegmodelljének szintjeit**!

1. Rendszerszint (System, Architectural Level)
2. Algoritmikus szint (Algorithmic Level)
3. Funkcionális blokkok szintje (Functional Block, Register Transfer Level)
4. Logikai szint (Logical Level)
5. Áramköri szint (Circuit Level)

4. Mondjon legalább két példát a **rendszerszint hardverrészegységeire**!

Rendszerszinten a számítógép hardverét félautomatikus fő részegységeivel írjuk le (például processzor, interfész, tároló stb.). A részegységeket funkcionálisan és protokolljaikkal jellemezzük.

5. Mivel jellemezzük a hardvert a rétegmodell **algoritmikus szintjén**?

Az algoritmikus szinten megadjuk az egyes részegységeket alkotó hardvermodulokat (például a processzor vezérlőegységből, ALU-ból, regisztertárolóból stb. épül fel), és az ezeket jellemző algoritmusokat, amelyek segítségével a hardvermodulok a bitsorozatot értelmezik.

6. Mit határozunk meg a **funkcionális blokkok szintjén**?

A funkcionális blokkok szintjén meghatározzuk a hardver modulok funkcionális blokkokból (például regiszterek, számlálók, összeadók stb.) való felépítését, valamint ezek műveletvégzését (például összeadás) és a köztük való adatátvitelt. Itt már figyelembe vesszük a jelek időbeli ütemezését (órjel).

7. A hardver melyik szintjén használjuk fel a **modellezés eszközeül a Boole-algebrát** és miért?

Logikai szinten a hardvert kapuáramkörökkel specifikáljuk, ennek eszköze a Boole-algebra. Strukturálisan ez kapuáramkörökből felépülő hardvert jelent.

8. Milyen **fizikai mennyiségeket** kezelünk **áramköri szinten**?

Az áramköri szinten a jeleket már folytonos mennyiségeknek tekintjük (feszültség, áramerősség stb.). A szintnek megfelelő strukturális építőelemek a tranzisztorok, kondenzátorok stb.

9. Az **Y-diagram** milyen három **nézőpontból** szemlélteti a hardver rétegmodelljét?

1. Strukturális felépítés
Ezzel írjuk le, hogy az egyes rétegekhez tartozó hardverelemek, milyen alacsonyabb szintű építőelemekből és milyen szerkezetben épülnek fel.
2. Viselkedési modell
Ez határozza meg az egyes rétegekhez tartozó építőelemek működésének modellezését.
3. Fizikai, geometriai jellemzők
Ezzel írjuk le az egyes szintekhez tartozó hardverelemek jellemző formáját, méretét stb.

10. Mit nevezünk **számítógép-generációnak**?

A számítógépes hardver fejlődésében ún. számítógép-generációkat szoktak megkülönböztetni. Az egyes számítógép-generációkat egymástól az alkalmazott alapáramkörök típusa, és a műveleti sebesség nagyságrendje szerint különböztetjük meg.

11. Ismertesse az **egyes számítógép-generációk jellemző hardver-építőelemeit**!

1. Első generáció
 - Hardware: Elektroncső, mágnesdob és szalag, relék
 - Software: Gépi kód, programkönyvtár
2. Második generáció
 - Hardware: Tranzistorok, mágneslemez, ferritmátrix
 - Software: Operációs rendszer, programnyelvek
3. Harmadik generáció
 - Hardware: Integrált áramkörök, félvezetőtárak, 16 kilobites chipek
 - Software: Többfelhasználós operációs rendszer, fejlett 3. generációs programnyelvek, adatbázis-kezelők
4. Negyedik generáció
 - Hardware: LSI áramkörök, mikroprocesszorok, 64 kilobites chipek, winchesterek
 - Software: Egységes, hordozható operációs rendszerek és adatbázis-kezelők, hálózati szoftverek, 4GL programnyelvek
5. Ötödik generáció
 - Hardware: VLSI, programozható áramkörök, 128 kilobites chipektől 1 Gigabites chipekig
 - Software: Világhálózati szoftverek, multimédia, mesterséges intelligencia

12. A számítógéprendszerek fejlődése során **milyen eszközökkel növelték a hardver teljesítményét**?

A számítógéprendszerek tervezői az alap-építőelemekbeli minőségi ugrások között rendszertechnikai megoldásokkal igyekeznek a számítógépek teljesítményét megnövelni. Például: utasításkészlet növelése, párhuzamosítás.

13. Miért van szükség **szabványos hardverleíró nyelvre**?

Mert egy szabványos "nyelven" biztosítja a hardver leírását, a magasabb szinteken az ember számára is áttekinthetővé teszi a hardver dokumentációját, az alsó szinteken lehetővé teszi a tervek átadását a gyártás CAD/CAM rendszereinek, és biztosítja a hardver szimulációs bevizsgálását.

14. Mi a VHDL?

A VHDL (Very High Speed Integrated Circuit Hardware Description Language) nevű hardverleíró nyelvet a 80-as évek közepén fejlesztették ki, melyet azóta nemzetközi szabványnak is elfogadtak.

15. Milyen lépései vannak a VHDL alapú hardverfejlesztésnek?

1. Feladatspecifikáció
2. Hardver leírás algoritmikus szinten
3. Hardver leírás funkcionális blokk szinten, VHDL nyelven
4. Logikai szintű kapuáramkör tervek (gyártófüggetlen)
5. Logikai szintű kapuáramkör tervek (gyártóspecifikus)
6. Integrált áramköri tervek

16. A hardverfejlesztés milyen lépéseiben vesz részt közvetlenül a tervező ember?

Az ember közvetlenül csak a hierarchia felső három szintjén kapcsolódik be a tervezőmunkába, ezt követően a tervezési feladatokat számítógépes rendszerekkel végeztetik el.

17. Milyen alapelemei vannak a VHDL nyelvnek, és ezekben mit kell megadni?

1. architektúra (architecture)
 - hardveregység funkciói, szerkezete
2. interfész (entity)
 - hardveregységek közötti kommunikáció (protokoll) jellemzői
3. konfiguráció (configuration)
 - architektúrák és interfészek egymáshoz rendelése

18. Milyen részekből épül fel egy hardvertervező szoftver?

1. Project manager: a tervező munka szervezését, irányítását és részben végzését segítő programokat tartalmazza.
2. Design manager: a tervezést segítő programokat tartalmazza.
3. Szimulátor, hardver-hibakereső: a megtervezett hardvereszközök szimulációval történő bevizsgálását biztosító programokat tartalmazza.
4. Építőelem könyvtárak (Resource Libraries): a gyakran előforduló hardverelemek standard terveit tartalmazza.

Számítógép-architektúrák

1. Mit nevezünk számítógéprendszer architektúrának?

Számítógéprendszer architektúrának a számítógép funkcionális felépítését, a részegységek kommunikációs kapcsolatait, valamint a rendszer specifikációjának együttesét nevezzük.

2. Mivel jellemzünk egy számítógéprendszer architektúrát?

Egy számítógép-architektúrát a részegységek és funkcióik, valamint ezek kapcsolatát meghatározó interfészprotokollok együttes leírásával jellemezhetünk.

3. Miért szükséges a chipek ciklikus működése?

Azért, mert a feszültségszintek csak bizonyos idő után állnak be a bináris 0-nak vagy 1-nek megfelelő értékre.

4. Hogyan számíthatjuk ki a **számítógép feladatvégrehajtási idejét**?

$F = C \times T \times U$, ahol C az egy utasításra eső átlagos ciklusszám, T az egy ciklushoz szükséges idő, U a feladatvégrehajtáshoz szükséges utasításszám.

5. Milyen módon növelhetjük meg egy számítógéprendszer teljesítményét a **feladatvégrehajtási idő csökkentésével**?

1. U csökkentése
 - hatékony programozás és fordítóprogram
2. T csökkentése
 - magasabb órajel frekvencia (áramköri fejlesztés)
3. C csökkentése
 - párhuzamosítás az utasításvégrehajtásban (architektúra fejlesztése)

6. Mi a **MIPS** és a **MFLOPS**?

MIPS - Million Instructions per Second

MFLOPS - Millions of Floating Point Operations per Second

7. Mit mutatnak a „**Benchmark**”-ok?

A számítógéprendszerek teljesítményéről összetettebb képet adnak a speciális teljesítménymérő programokkal előállított mutatók, a „Benchmark”-ok.

8. Milyen **szempontok szerint csoportosíthatjuk a számítógéprendszereket**?

1. Teljesítmény szerint: mikro- (kis-), közép- és nagyszámítógépek
2. Az utasítás- és adatfolyamok száma szerint: SISD (egy utasítás-, egy adatfolyam), SIMD (egy utasítás-, több adatfolyam), MISD (több utasítás-, egy adatfolyam) és MIMD (több utasítás-, több adatfolyam) gépek
3. Az utasításkészlet szerint: komplex és egyszerűsített utasításkészletű gépek (CISC és RISC)
4. A számítógép működési elve szerint: Neumann és nem Neumann elvű architektúrák
5. Az egy időben kiszolgált felhasználók száma és a kiszolgálás időbelisége szerint.

9. Mi jellemzi a **nagyszámítógépeket**?

1. nagy műveleti sebesség
2. nagy tárolókapacitás
3. speciális működési feltételek (kondicionálás, pormentesség stb.)
4. nagy megbízhatóságú működés (pl. 24 órás folyamatos, leállás nélküli üzem)

10. Mutassa be a **hibatűrő architektúrák fontosabb jellemzőit**!

1. a paritásellenőrzés mellett (helyett) hibajavító kódok alkalmazása
2. rekonfigurációs képesség, azaz a meghibásodott hardveregységeket a rendszer automatikusan, a folyamatos működés leállítása nélkül felismeri és izolálja és ezek nélkül működik tovább
3. többszörözött hardveregységek
4. automatikus tömörítés és titkosítás a háttértárolókon
5. szervizprocesszor, távfelügyelet

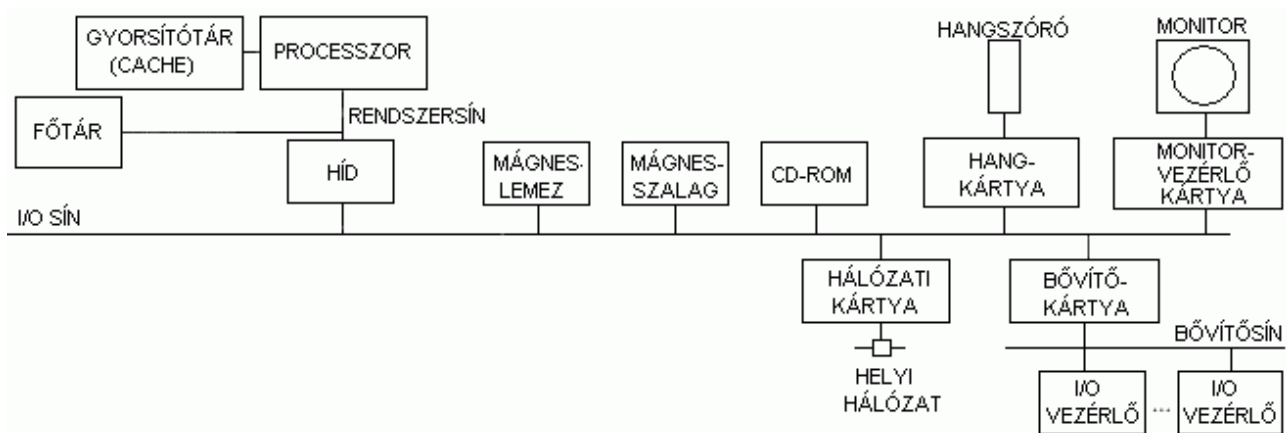
11. Jellemezze a **közepes teljesítményű számítógépeket**!

A közepes teljesítményű számítógépek (minicomputer, workstation) átmenetet képeznek a nagy- és a mikroszámítógépek között. Jellemző felhasználási területük: helyi hálózatok szerverei, adatbázis- és médiaszerverek.

12. Mit nevezünk kis vagy **mikroszámítógépnek**?

Ezeket a gépeket általában hálózatok munkaállomásaként vagy önállóan használják, a mikrogépek processzora egyetlen áramköri egységet alkot.

13. Mutassa be a **mikroszámítógépek elvi felépítését** rajzban!



14. Mi a **NETPC**?

A mikroszámítógépek speciális változata a NETPC. Ennek alap gondolata, hogy helyileg nincs szükség teljes körű PC-funkciókra, mert ezeket hálózati szolgáltatásokkal lehet helyettesíteni.

15. Definiálja az **adat- és utasításfolyam** fogalmát!

Utasításfolyam: Az utasítások egymásutáni sorozata, amelyeket egy program lefuttatása során hajt végre a számítógép.

Adatfolyam: Az utasításfolyam utasításai mindig meghatározott adatokra hivatkoznak, amelyekkel a műveleteket el kell végezni. Ezek egymásutánosságát, milyen sorrendben rendelkezésre kell állniuk, nevezzük adatfolyamnak.

16. Mit értünk **SISD** architektúra alatt? És **SIMD** alatt?

A **SISD** (Single Instruction Stream Single Data Stream) típusú számítógép egyetlen utasításfolyammal egyetlen adatfolyamot dolgoz fel.

A **SIMD** (Single Instruction Stream Multiple Data Stream) típusú számítógép egyetlen utasításfolyammal többszörös adatfolyamot dolgoz fel. Ezek a számítógépek több párhuzamos, egy idejű működésre képes műveletvégző egységet (ALU) tartalmaznak. Vektorműveleteket képesek végrehajtani, gépi utasítás szinten.

17. Mit értünk **MISD** architektúra alatt? És **MIMD** alatt?

A **MISD** (Multiple Instruction Stream Single Data Stream) típusú számítógép több utasításfolyammal egyetlen adatfolyamot dolgoz fel, gyakorlatban ilyen gépek nem léteznek.

A **MIMD** (Multiple Instruction Stream Multiple Data Stream) számítógéprendszerek több utasításfolyammal több adatfolyamot dolgoznak fel, ebbe a kategóriába tartoznak a multiprocesszoros gépek.

18. Sorolja fel a **Neumann elveket**!

1. számítógép működését tárolt program vezérli
2. a vezérlés control-flow (vezérlésáramlásos) rendszerű, ami azt jelenti, hogy a gép a program utasításait egymás után sorban hajtja végre
3. a gép belső tárolójában a program utasításai és a végrehajtásukhoz szükséges adatok egyaránt megtalálhatók
4. az aritmetikai és logikai műveletek (programutasítások) végrehajtását önálló részegység (ALU = Arithmetic Logic Unit) végzi
5. az adatok és programok beolvasására, tárolására és megjelenítésére önálló egységek (perifériák) szolgálnak

19. Milyen módszerei vannak a **számítógépek nem strukturális gyorsításának**?

1. Az órajel frekvenciájának növelése
2. A program optimalizált fordítása

20. Milyen módszerei vannak a **számítógépek strukturális gyorsításának**?

1. Párhuzamosítás a CPU-n belül
 - vektorszámítógépek
 - pipeline-feldolgozás
 - szuperskalár processzorok
2. Párhuzamosítás a CPU-n kívül
 - társprocesszorok
 - multiprocesszoros architektúrák
3. Nem hagyományos elven működő számítógépek
 - neurális hálók

21. Mit nevezünk **CISC** processzornak?

Complex Instruction Set Computer - nagy számú utasítást tartalmazó utasításkészlettel rendelkeznek, az utasítások szerkezete bonyolult, többfajta memóriacímzési módot tesznek lehetővé, az utasításvégrehajtás mikroprogramvezérelt.

22. Mit nevezünk **RISC** processzornak?

Reduced Instruction Set Computer - utasításkészlete csökkentett, egyszerűsített, a memóriáhozáférés csak két utasítással: memóriából való adatbetöltés (LOAD) és memóriába való adatírás (STORE) történhet, a műveleti vezérlés huzalozott (azaz beépített áramkörökkel végrehajtott) vagy horizontális (egyszerűsített) mikroprogram-vezérelt.

23. **Hasonlítsa össze a RISC és CISC processzorok jellemzőit!**

CISC processzorok

Összetett utasítások, amelyek végrehajtása több gépi ciklust igényel. Bármely, erre alkalmas utasítás igénybe veheti a tárolót.
A futószalag (pipelining) feldolgozás kismértékű.
Változó hosszúságú utasítások.
Sokféle utasítás és címzési mód.
Bonyolult mikroprogram, egyszerű fordítóprogram.
Kis számú regiszter.
Tárolóvédelem hardverúton.

RISC processzorok

Egyszerű utasítások, amelyek végrehajtása 1 gépi ciklust igényel. Csak a LOAD/STORE utasítások fordulhatnak a memóriához.
Erőteljes futószalag (pipelining) feldolgozás.
Rögzített utasításhossz.
Kevés utasítás és címzési mód.
Bonyolult fordítóprogram, egyszerű mikroprogram.
Nagy méretű regisztertár.
Tárolóvédelem szoftver segítségével.

24. Mit nevezünk **vektorszámítógépnek**?

Olyan SIMD (egyszeres utasításfolyam, többszörös adatfolyam) architektúrájú gépek, amelyek a gépi utasítás szintjén képesek vektor- illetve mátrixműveleteket végrehajtani.

25. Mi a **pipelining** lényege?

A pipelining átlapoltt utasítás-végrehajtást magyarra fordítva csővonal, adatcsatorna vagy futószalag feldolgozásnak nevezhetjük. A gépi utasítások elemi lépései (előkészítés, dekódolás, operandusok címszámítása stb.) különböző hardver-erőforrásokat igényelnek. Ezért, ha egy utasításban egy elemi lépést végrehajtottunk és az ehhez szükséges hardveregység felszabadul, akkor ezt igénybe vehetjük egy következő utasítás elemi lépésének végrehajtására.

26. Mit nevezünk **ILP processzornak**?

A pipeline utasítás-feldolgozást alkalmazó processzorokat utasításszinten párhuzamos működésű (Instruction Level Parallel), vagy ILP processzoroknak nevezzük.

27. Definiálja a **szuperskalár processzorokat**!

A processzorokat szuperskalár processzoroknak nevezzük, ha egy gépi ciklus alatt esetenként több utasítást is képesek végrehajtani.

28. Mit jelent a **TLP-feldolgozásmód**, és hogy valósítják meg?

Az egy folyamathoz tartozó szálak egyidejű végrehajtására képes feldolgozásmódot alkalmazó processzorokat szálszinten párhuzamosaknak, rövidítve TLP (Thread Level Parallel) processzoroknak nevezzük. A végrehajtóegységek minél jobb terhelése érdekében a processzor minden ciklusban akármelyik szálból kiválaszthat végrehajtandó utasítást. A folyamatok szálakra bontásához szoftvertámogatás is szükséges.

29. Jellemezze a CPU-n kívüli **társprocesszorokat**!

A társprocesszorok konkrét feladatokra, önálló működésre alkalmas processzorok, amikkel bővítjük az architektúrát a processzor tehermentesítése érdekében.

30. Mondjon legalább **három példát társprocesszor** alkalmazására mikroszámítógépeknél!

1. műveletvégző társprocesszor
2. 3D gyorsító társprocesszor
3. hangfeldolgozó társprocesszor

31. Milyen **igények** okozzák a multiprocesszoros számítógépek fejlődését?

Hibatűrő architektúrák iránti igény, skálázható számítógéprendszerek iránti igény.

32. Mit nevezünk **multiprocesszoros** számítógépnek?

A multiprocesszoros számítógépek lényege, hogy több vezérlésre és műveletvégrehajtásra önállóan alkalmas processzorral működnek.

33. Multiprocesszoros architektúrában mit jelent a **durva és a finom szemcsés** feladatkiosztás?

1. Az egyes processzorok nagyobb, önálló feladatokat oldanak meg (kis mértékű kommunikáció), ez a „durva szemcsés” feladatkiosztás.
2. Az egyes processzorok kisebb, egyszerű feladatokat oldanak (erős kommunikációs igény), ez a „finom szemcsés” feladatkiosztás.

34. Milyen változatok vannak multiprocesszoros rendszerekben az **erőforráshasználat megszerzésére**?

1. Közös erőforrások (pl. memória) esetén meg kell határozni a processzorok és az erőforrások kiosztási eljárását. A közös memóriahasználat adatkoherencia problémákat vet fel (ugyanazt az adatelemet több processzor is módosíthatja).
2. Részben vagy teljesen saját erőforrásokkal rendelkező processzorok esetében a processzorok közötti intenzív kommunikációt és szinkronizálás működését kell megszervezni.

35. A **processzorok közötti kommunikáció megszerzéséhez** milyen eljárásokat kell meghatározni?

1. Ütemezés (aszinkron, szinkron)
2. Kapcsolatfelépítési algoritmus
3. A kommunikációs kapcsolatok vezérlése (központi, decentralis)
4. A kommunikációs hálózat térbeli kiépítése vagy topológiája (statikus, dinamikus)

36. Mi a **transzputer**?

A transzputer egy egyetlen chipen megvalósított RISC alapú processzor, amelyhez egy memóriamodul is tartozik. Rendelkezik egy DMA-vezérlővel (Direct Memory Access), amely képessé teszi más transzputerekkel való kapcsolattartásra. Ezért megfelelő számú transzputer, „építőkocka-szerű” összekapcsolásával igen nagy teljesítményre képes szuperszámítógépek építhetők fel.

37. Mi az **SMP** architektúra?

Lényegében a transzputer alapgondolatát fejlesztette tovább az SMP (Symmetric Multiprocessors) típusú, napjainkban általánosan elterjedt multiprocesszoros architektúra. Ez azt jelenti, hogy több, azonos, sorozatban gyártott processzorból épülnek fel a szuperszámítógépek. Az SMP architektúrájú rendszerekben mindegyik processzor egyformán elérheti az összes memóriamodult és I/O eszközt.

38. Miben különbözik az **UMA** és a **NUMA** rendszer?

Az SMP architektúrájú multiprocesszoros rendszerek működhetnek egységes memóriahasználattal (UMA = Uniform Memory Access), ekkor a processzorokat és a közös memóriát egy sínrendszer vagy egy kapcsolóhálózat köti össze.

A NUMA (NonUniform Memory Access) multiprocesszoros rendszerben minden processzorhoz egy lokális memóriamodul is tartozik, amelyet a processzor a többi memóriánál jóval gyorsabban elérhet.

39. Mit jelent a **skálázhatóság**?

Azt jelenti, hogy az igényelt számítási teljesítménynek megfelelően alakítják ki a rendszert "építőkockákból", és ez a konfiguráció bármikor a folyamatos működés leállítása nélkül továbbépíthető.

40. Milyen **alap építőelemek** találhatók egy **SMP-architektúrában**?

1. alap processzormodul
2. alap I/O modul
3. háttértároló-modul
4. grafikus bővítő modul
5. kommunikációs modul
6. tápegységmodul

41. Mit jelent az **automatikus hibafelismerés**?

A rendszer képes az automatikus hibafelismerésre, és a hibás építőelem vagy partíció lekapcsolására, anélkül, hogy ez a teljes rendszer leállítását eredményezné.

42. Mit neveznek **multiszámítógépnek**?

A több ezer processzorból álló rendszerek kialakítására az üzenetátadási architektúra a megfelelő, amelyek esetében az önálló számítógépekből álló csomópontokat egy nagy teljesítményű hálózat köti össze. Ezt az architektúrát multiszámítógépnek (multicomputer) is nevezik.

43. Melyek a legfontosabb **jellemzői** a **neurális hálózatnak**, és milyen tipikus felhasználói területeik vannak?

A mesterséges neuronhálózatokon alapuló számítógépek az emberhez hasonlóan taníthatóak, képesek nagyfokú általánosításra (absztrakcióra) és asszociációra.

Alkalmazásuk:

1. alakfelismerés
2. képfeldolgozás
3. hangfelismerés

44. Hogyan állítja elő egy sejt a **neurális hálózatban** az **inputból** az **outputot**?

A neurális hálózatok lényege a neuronok (sejtek) közötti kapcsolat, melyet az úgynevezett szinaptikus súlyokkal lehet vezérelni. Ha egy sejt több, más neurontól kap input értéket, akkor mindegyik input érték egy szinaptikus súllyal lesz megszorozva. Az így kapott szorzatokat összeadja a fogadó neuron, majd az értéket egy átviteli függvény szerint áttranszformálja, és az így kapott érték lesz a neuronsejt outputja.

45. Hogyan „**emlékezik**” a **neurális háló**?

A neurális hálózatokban a tanulás során a neuronok közötti kapcsolat erőssége, vagyis a szinaptikus súly változik egy tanulási szabály alapján, azaz a sejhálózatban a memóriát lényegében a szinaptikus súlyok értékével modellezzik.

46. Milyen **osztályozási** szempontjai vannak a **számítógéprendszerek üzemmódjainak**?

1. az egy időben kiszolgált felhasználók száma
2. az „egy időben” futtatható programfolyamatok száma
3. a felhasználói igények kiszolgálásának időbelisége
4. a rendszer térbeli felépítése

47. Mit értünk **egy-** és **többfelhasználós üzemmód** alatt? Mi a feltétele a többfelhasználós üzemmódnak?

Egyfelhasználós üzemmódban egyidejűleg csak egy felhasználói program van a gép memóriájában.

A többfelhasználós üzemmódú számítógépes rendszerek egyidejűleg több felhasználó kiszolgálására képesek. Ehhez arra van szükség, hogy a számítógép erőforrásait (processzor, memória stb.) valamilyen stratégia (algoritmus) szerint az operációs rendszer megossza az egyes feladatok között.

48. Mit nevezünk **multiprogramozásos** rendszernek?

A multiprogramozásos rendszerekben egy időben több programfolyamat (job, task, process) kerül betöltésre a számítógép memóriájába, amelyek valamilyen eljárás szerint osztozkodnak a programfutáshoz szükséges erőforrásokon (memória, I/O eszközök használata stb.).

49. Milyen lényeges **feltétele** van a **multiprogramozásnak**?

A multiprogramozott üzemmódnak két lényeges feltétele, hogy a memória a programfolyamatoknak megfelelően részekre (partíciók) legyen osztva, és az I/O műveletet végrehajtó egységek önálló vezérléssel rendelkezzenek, azaz a processzortól függetlenül is képesek legyenek a feladatvégrehajtásra.

50. Definiálja a **prioritásos** és az **időosztásos** multiprogramozást!

Prioritásos elv: az egyes feladatoknak megfelelő programfolyamatoknak prioritása van, azaz fontosságuknak megfelelő sorrendben kapják meg a számítógép erőforrásait (processzor, memória, I/O vezérlések).

Időosztásos elv: az egyes feladatoknak megfelelő programfolyamatokhoz ún. „időszeletek” kerülnek hozzárendelésre, amelyek meghatározzák, hogy az adott feladat mennyi időre veheti igénybe a processzort. Az időosztás elvén megszervezett multiprogramozásos üzemmódot időosztásosnak (time-sharing) nevezik. Ezen az elven működik az összes korszerű operációs rendszer.

51. Mit jelent a **batch processing** és a **remote batch**?

Kötegetelt feldolgozás (Batch Processing) esetén a programfolyamathoz szükséges összes adatot először összegyűjtik, és számítógépes feldolgozásukat időszakosan hajtják végre (például naponta, havonta, évente meghatározott időpontban). Erre példát a mikroszámítógépeknél a *.BAT kiterjesztésű fájlok végrehajtása során találhatunk. Amennyiben az adatkommunikáció kötegetelt feldolgozásnál távadat-feldolgozással történik, akkor ezt remote batch-nek nevezzük.

52. Mikor interaktív a számítógépes feldolgozás?

Interaktív feldolgozás (Interactiv Processing) esetén a felhasználó állandó és folyamatos kapcsolatban áll a feladatot végrehajtó programfolyamattal. Ennek előnye, hogy a feladatot előre nem kell részleteiben megtervezni, mivel a felhasználó a feladat végrehajtását maga vezérli, szükség esetén lépésenként módosíthatja a részfeladatok sorrendjét.

53. Határozza meg a **dialógus üzemmód** fogalmát!

Dialógus üzemmód esetén a felhasználó kérdez-felelek módon, lépésenként folyamatosan kommunikál a programfolyamattal. Ennek tipikus megvalósítása a terminál üzemmód, amikor a felhasználó ember egy képernyőn közli a rendszerrel a parancsokat és adatokat, és a rendszer az adatokat szintén egy képernyőn jeleníti meg.

54. Mi jellemzi a **folymatvezérelt üzemmódot** és milyen fontosabb felhasználási területei vannak?

Folyamatvezérelt üzemmód (Process Control) esetben a „felhasználó” egy gépi folyamat, amit a számítógép vezérel. Ilyenek lehetnek például az ipari folyamatok irányítását, forgalmi rendszerek vezérlését stb. végző számítógépes rendszerek, melyek különböző mérőműszerektől, érzékelőktől érkező jeleket fogadnak és dolgoznak fel.

55. Definiálja a **real-time rendszer** fogalmát!

A dialógus és a folyamatvezérelt üzemmód egyaránt lehet valós idejű (real-time) feldolgozás, ami azt jelenti, hogy a számítógépes rendszer meghatározott kritikus időn belül azonnal feldolgozza a fogadott adatokat.

56. Mi a különbség a **centralizált** és az **elosztott számítógéprendszer** között?

Centralizált rendszerben csak egy központi vezérlés és feladat-végrehajtó egység található.

Elosztott számítógéprendszerben több, térben esetleg távoli, önálló vezérléssel rendelkező és feladatvégrehajtásra képes számítógép található, amelyeket egy hálózat köt össze, és adatátviteli vonalakon (adatbuszok, telefonvonalak stb.) tartanak kapcsolatot egymással. Az egyes részfeladatokat ellátó számítógépek (Slave Computer) együttműködését általában egy kitüntetett számítógép (Master Computer) vezérli.

57. Mit jelent az **on-line** és az **off-line** kapcsolat?

On-line, ha a két eszköz fizikailag úgy összekapcsolt, hogy közöttük aktív vezérlési és/vagy adatátviteli kapcsolat áll fenn.

Off-line, ha két eszköz között nincs aktív vezérlési és/vagy adatátviteli kapcsolat.

A processzor felépítése és működése

1. Sorolja fel a **processzor** legfontosabb architektúrais **építőelemeit**!

1. vezérlőegység (CU = Control Unit)
2. aritmetikai és logikai egység (ALU = Arithmetic Logic Unit)
3. regiszterkészlet
4. buszillesztő egység (BIU = Bus Interface Unit)
5. címszámító és védelmi egység (AU = Address Unit)
6. belső gyorsítótár (L1 cache)
7. az előző négy részegység kommunikációját biztosító eszközök (mikroszámítógépeknél a belső sínrendszer)

2. Mi a feladata a **vezérlőegységnek**, és mit tartalmaz a **PC** és az **IR**?

A vezérlőegységnek a feladata a programban lévő utasítások alapján a teljes számítógép részegységeinek (aritmetikai egység, memória, kommunikációs eszközök, háttértár és perifériavezérlések) irányítása, összehangolása.

PC = Program Counter, amely a soron következő utasítás tárolóbeli címét tartalmazza.

IR = Instruction Register, amely a memóriából kiolvasott utasítást tárolja. Az ebben található műveleti kód alapján a vezérlőegység meghatározza az elvégzendő műveletet (dekódolás).

3. Mi a feladata az **aritmetikai egységnek**, és mit tartalmaz az **AC** és a **FLAG regiszter**?

Az aritmetikai egység az utasításkódokban előírt aritmetikai és logikai (pl. logikai ÉS, VAGY stb.) műveleteket hajtja végre. Az aritmetikai egységben általában a következő két regiszter mindig megtalálható:

AC = Accumulator Register, amely a műveletvégrehajtásnál az adatok (operandus) átmeneti tárolására szolgál.

FLAG regiszter = állapotjelző regiszter, amelyben a végrehajtott utasítás következtében megváltozott állapotok kerülnek bitenként kódolásra.

4. Mi a feladata a **buszinterfész** (BIU) a **címszámító** (AU) egységnek?

A buszinterfész egység biztosítja a processzor kapcsolódását a külső sínrendszerhez.

A címszámító és védelmi egység feladata a programutasításokban található címek leképezése a főtár fizikai címekre és a tárolóvédelmi hibák felismerése.

5. Mire szolgál a **belső sínrendszer** és a **belső gyorsítótár**?

A belső sínrendszer a CPU-n belüli adatforgalmat lebonyolító áramkörök összessége.

A belső gyorsítótár (L1 cache) a főtárból kiolvasott utasítások és adatok átmeneti tárolására szolgál.

6. Definiálja a Pentium processzorcsalád **valós üzemmódját**, fejtse ki, hogy miért van erre szükség!

Valós üzemmódban a processzor úgy működik, mint egy régi (PC-XT) 8086-os processzor. Erre azért van szükség, hogy a felhasználók változatlan formában képesek legyenek „rég”, megszokott DOS-os programjaik futtatására.

7. Definiálja a Pentium processzorcsalád **védett üzemmódját**!

Védett üzemmódban kihasználhatók a 32-bites architektúra összes lehetőségei, ezért a teljesítmény ekkor a legnagyobb. Az üzemmód onnan kapta a nevét, hogy ebben az állapotban a tárolóvédelmi rendszer bekapcsolásra kerül.

8. Mit jelent a Pentium processzorcsalád **védett valós** és **rendszermenedzselő üzemmódja**?

Védett valós üzemmódban a processzor a 8086-os processzort csak egy taszkban emulálja.

A rendszermenedzselő üzemmód a processzor energiatakarékos működési módja.

9. Hogyan csoportosíthatók a **processzor regiszterei**?

1. Rendszerregiszterek, amelyek a felhasználói programok számára nem „láthatók”, nem elérhetők. Erre például szolgál az IR utasítás regiszter.
2. Speciális célú regiszterek, amelyek a felhasználói programokban csak meghatározott utasításokban szerepelhetnek. Erre példa a flag.
3. Általános célú regiszterek, amelyeket a felhasználói programok utasításaiban korlátozás nélkül használhatók. Erre példa az akkumulátor regiszter.

10. Mit értünk a **gépi utasítás szerkezete** alatt?

Utasításszerkezet alatt az elemi szintű (gépi kódú), a processzor által közvetlenül értelmezhető utasítások felépítését értjük.

11. Milyen **részekből** épül fel egy **gépi utasítás**?

1. műveleti kód (opcode), amely meghatározza a processzor számára, hogy milyen műveletet kell végrehajtania
2. operandus hivatkozások, amelyek azt adják meg a processzor számára, hogy hol található, illetve milyen adatokkal kell a műveletet végrehajtani
3. kiegészítő, módosító rész, amelyben olyan kiegészítő információkat adhatunk meg a processzor számára, amelyek a műveleti jelrész (kód), illetve az operandus hivatkozások teljes körű (pontos) értelmezéséhez szükségesek

12. Definiálja az **utasításkészlet** fogalmát!

Egy processzor utasításkészlete alatt azoknak az elemi szintű gépi kódú utasításoknak az összességét értjük, amelyek végrehajtására a processzor a legalsó, hardver szinten képes.

13. Sorolja fel az **utasításkészlet fontosabb utasítástípusait**, és ismertesse az adatátviteli utasításokat!

1. adatátviteli, adatmozgató utasítások
2. műveletvégrehajtó utasítások
3. vezérlő utasítások

Adatátviteli utasítások: MOVE, IN, OUT, POP, PUSH

14. Milyen műveleteket lehet elvégezni a **műveleti utasításokkal**?

ADD, SUB, MUL, DIV, AND, OR, XOR, SLL, BTC

15. Mikor és milyen célból kerülnek beállításra az állapot (flag) regiszter fontosabb jelzőbitjei (**carry, zero, sign, overflow**)?

1. átvitel (carry) jelzőbit 1-esre lesz beállítva, ha a műveleti eredményben, a legmagasabb helyiértéken átvitel történik
2. nulla (zero) jelzőbit 1-esre lesz beállítva, ha a művelet eredménye 0
3. előjel (sign) jelzőbit 1-esre lesz beállítva, ha a művelet eredménye negatív
4. a túlcsordulás (overflow) jelzőbit 1-esre lesz beállítva, ha a művelet eredménye nagyobb mint a tárolható legnagyobb érték

16. Ismertesse a **vezérlő utasításokat**!

CALL, RET, LOOP, JNE, JMP

17. Milyen **adattípusokat dolgoz fel** az **ALU**?

1. fixpontos számok
2. lebegőpontos számok
3. binárisan kódolt decimális (BCD) számok
4. multimédiás (MMX) adatok
5. karakteres mezők
6. bitmezők

18. Milyen **szempontok** szerint értékeljük a **processzorok tárolókezelő rendszerét**?

A tárolókezelés értékelésénél a legfontosabb a címzési módok elemzése. Itt vizsgálni kell a relatív (bázis), közvetett (indirekt), közvetlen adat és indexelt címzés adta lehetőségeket, valamint a virtuális címekből a fizikai címek kiszámításának módszerét, valamint az ehhez felhasznált eszközöket (regiszterek, memóriatáblázatok) és a címszámítás hardvermegoldását.

19. Definiálja a **megszakítás** és a **kivétel** fogalmát!

Megszakítás (interrupt), amikor a program utasításainak átmeneti felfüggesztését a processzortól független, külső esemény idézi elő.

Kivétel (exception), amikor a programot a programutasítás végrehajtása alatt a processzoron belül fellépő esemény miatt kell megszakítani.

20. Milyen **ábrázolási formák** szerint csoportosíthatjuk a **numerikus** adatokat?

A numerikus adatok ábrázolásának két legfontosabb típusa az előjeles és a nem előjeles forma.

Előjeles számábrázolást alkalmazunk a szokásos matematikai számítások végrehajtásához. Ezek legfontosabb formái:

1. a fixpontos számábrázolás
2. a lebegőpontos számábrázolás
3. a decimálisan kódolt bináris számábrázolás

21. A **fixpontos számok** milyen formában kerülnek **tárolásra** a számítógépen?

A fixpontos számokon végrehajtott műveleteknél az operandusok általában szószervezésben kerülnek tárolásra. A szóban található bitsorozat kettes számrendszerben megadja az adott szám tényleges értékét, a legnagyobb helyiértékű bit pedig a szám előjelét határozza meg (0 = pozitív, 1 = negatív).

22. Ismertesse, hogy a **fixpontos alpműveleteket** milyen módon lehet **visszavezetni összeadásra és shiftelésre**!

Két fixpontos szám kivonását visszavezethetjük az összeadásra, ha a negatív szám tárolására a kettes komplementkódot használjuk. Emiatt az összeadás és kivonás azonos algoritmussal és azonos műveletvégrehajtó áramkörökkel elvégezhető.

Két fixpontos szám szorzását és osztását viszont összeadások, illetve kivonások sorozatára, valamint balra és jobbra történő 1 bites léptetésekre (ez a shiftelés ugyanis 2-vel való szorzást, illetve osztást jelent) vezethetjük vissza.

23. Hogyan **ábrázoljuk** és **milyen esetekben használjuk** a **lebegőpontos számokat**?

Lebegőpontos szám ábrázolásához két fixpontos szám – egyik a karakterisztika, másik a mantissa – tárolását kell megoldanunk, amelyeket a számítógépen történő művelet-végrehajtáskor együtt vesszük figyelembe.

A lebegőpontos számábrázolást általában olyan számításigényes algoritmusoknál használjuk, amelyeknél a szükséges számtartomány (a legkisebb és a legnagyobb érték) eltérése jelentős mértékű.

24. Mit tartalmaz az IEEE 754-es szabvány?

A szabvány meghatározza a lebegőpontos számábrázolás normalizált formáját.

1. A mantissza előjele (fixpontos számként) 0, ha a szám pozitív és 1, ha negatív.
2. A mantisszában lévő "a" fixpontos szám 1-re normalizáltan értendő, azaz 1.a értékű.

25. Az IEEE 754-es szabvány szerint mit jelent a denormalizált, a végtelen szám, a nulla és a „nem szám”?

A denormalizált szám az igen kis számértékek számítógépes kezelését segíti.

A túlcsordulásokat a „végtelen” számmal kezelhetjük.

A nulla ($\pm$) ábrázolása konvencionális.

Az úgynevezett „nem számok” azt a célt szolgálják, hogy programunk akkor se álljon le, ha az elvégzett művelet matematikailag értelmezhetetlen.

26. Hogyan lehet visszavezetni a lebegőpontos műveleteket fixpontos műveletekre?

A lebegőpontos műveletek végrehajtását – mivel ez esetben két fixpontos számmal, a mantisszával és karakterisztikával kell végrehajtani aritmetikai műveleteket – szintén vissza tudjuk vezetni fixpontos összeadásra.

27. Mit nevezünk BCD kódnak és hogyan hajtjuk végre a műveleteket BCD kódú számokkal?

A BCD (Binary Coded Decimal) kódban egy decimális számjegyet bináris értékének megfelelően tárolunk. A BCD kódú műveletvégzésnél a 4 bittel kódolt decimális számjegyeket számjegyenként – azaz 4 bites egységekben – a tízes számrendszerben adjuk össze.

28. Ismertesse az MMX aritmetikai műveleteket!

SIMD típusú gépi utasítások a vektor adattípusok kezelésére, telített vagy körbefűzéses formában.

29. Indokolja meg, hogy miért hajtható végre az összes aritmetikai és logikai művelet kapuáramkörökkel!

A szokásos Boole-algebrai műveleteket végző gépi kódú utasításokat (NEM, ÉS, VAGY, AZONOSSÁG stb.) áramköri szinten végre lehet hajtani a KIZÁRÓ VAGY és az ÉS logikai műveleteket modellező áramkörökkel (1 bites összeadó).

30. Milyen áramkörökből épül fel az ALU?

1. összeadó egységekből, amelyek alapeleme két 1 bites fixpontos adatelem összeadását végrehajtó áramkör
2. léptető áramkörökből, amelyek az operandusokat tároló hardverregiszterek jobbra vagy balra léptetését hajtják végre
3. logikai műveleteket végrehajtó áramkörökből
4. az operandusokat, illetve eredményeket tároló átmeneti tárolókból, amelyek közül legfontosabb az aritmetikai egység akkumulátor regisztere

31. A **gépi utasítást** milyen **elemi lépésekben** hajtja végre a processzor?

1. Utasítás előkészítés vagy lehívás. Ebben a fázisban a processzor az utasításszámláló (PC) tartalma alapján kikeresi a főtárból az utasítást és átviszi a vezérlőegység utasításregiszterébe (IR).
2. Az utasításszámláló regiszter tartalmának növelése. Az utasításszámláló regiszterhez hardveráramköri úton hozzáadásra kerül az aktuálisan végrehajtandó utasítás hossza, így az most már a következő végrehajtandó utasítás címét fogja tartalmazni.
3. Az utasítás dekódolása, műveleti kód és az utasításszerkezet értelmezése, az utasításban lévő operandusok címének számítása.
4. A művelet végrehajtásához szükséges adatok kiolvasása a főtárból, előkészítése a végrehajtáshoz.
5. A műveleti kód alapján értelmezett művelet végrehajtása az előkészített operandusokkal. (Vezérlésátadó utasítás esetén ebben a lépésben a következő utasítás címét írja be a processzor az utasításszámláló regiszterbe.)
6. A művelet eredményét a processzor beírja az utasításban előírt tárolóhelyre.

32. Mit nevezünk **műveleti vezérlésnek**?

Műveleti vezérlésnek a gépi utasítások elemi lépéseinek ciklusonkénti végrehajtásához a számítógép összes, a művelet-végrehajtáshoz szükséges hardverrészegységének a gépi utasítás alapján történő összehangolt irányítását nevezzük.

33. Miért van szükség a műveleti vezérléshez **vezérlő mezőkre** és **vonalakra**?

A vezérlési pontokhoz vezérlő mezők tartoznak, és ezek dekódolásával, a vezérlővonalakon keresztül jut el az utasítás műveleti kódjának megfelelő vezérlési információ a vezérlési pontokhoz.

34. Hogyan **csoportosíthatók** a vezérlőegység **vezérlőjelei**?

1. A processzor belső vezérlőjelei, amelyek a processzoron belüli részegységek működését irányítják, például az aritmetikai egység és a processzor regiszterei közötti adatátvitelt.
2. A processzoron kívüli egységek irányítását szolgáló külső vezérlőjelek, amelyek a processzor és a memória, a processzor és az input/output eszközök közötti adatátvitelt, illetve a megszakítás-kezelést és a sínvezérlést irányítják.

35. Mit nevezünk **huzalozott** és **mikroprogramozott** műveleti vezérlésnek?

1. a vezérlési pontokat áramkörökkel állítjuk be (huzalozott vezérlés)
2. a vezérlési pontokat a gépi utasítás kódja alapján egy ROM tárolóból kiolvasott mikroprogram utasításai állítják be lépésről-lépésre (mikroprogramozott vezérlés)

36. Mi a **horizontális mikroprogramozás** lényege?

A horizontális mikroutasítás egy vezérlési mezőt és a következő mikroutasítás címét tartalmazza (ha van ilyen). A vezérlési mező minden egyes bitje az áramkörök egy-egy vezérlési pontját állítja be.

37. Mi a **vertikális mikroprogramozás** lényege?

A vertikális mikroutasítás lényegében úgy épül fel, mint egy gépi kódú utasítás. Általában mikroutasítások sorozata kerül sorosan végrehajtásra.

38. Milyen processzorokra igaz az a kijelentés, hogy „számítógép működik a számítógépen belül”, és ez mit jelent?

CISC processzorok esetében: a művelet-végrehajtás elemi lépéseit egy mikroprogramozott vezérlőegység irányítja a tárolt mikroprogram alapján.

39. Milyen összefüggés van a mikroprogramvezérlés és a processzor architektúrája között?

A mikroprogramvezérlés és a processzor architektúrája között szoros összefüggés van:

- a horizontális mikroprogramvezérlés (ennek szélsőséges esete a huzalozott műveleti vezérlés) a RISC processzorokra jellemző
- a vertikális mikroprogram-vezérlést viszont a CISC processzorokban alkalmazzák

40. Hasonlítsa össze a soros és a pipeline utasításfeldolgozást!

Soros végrehajtás esetén az utasításokat sorban hajtják végre, ha valamilyen műveletre várni kell, akkor a működés leáll. Pipeline esetén futószalagszerűen hajtódnak végre az utasítások, nem kell megvárni az egymástól független műveletek eredményét.

41. Mutassa be a pipelining működés során fellépő problémákat!

1. Az utasítások elemi fázisainak végrehajtásához szükséges idő igen eltérő lehet (például ha az operandus egy regiszterben található, akkor innen kb. tízszer gyorsabban lehívható, mint a főtár memóriájából).
2. Az utasítás soros végrehajtását a vezérlésátadó utasítások megzavarhatják, mivel ekkor nem a soron következő utasításokat kell betölteni a „futószalagra”. Ezek az ún. Vezérlésfüggőségek.
3. A megszakítások, kivételek kezelése is megszakíthatja a futószalag folyamatos működtetését.
4. Az utasításvégrehajtás során sokszor előfordul, hogy egy utasítás a megelőző utasítás eredményadatára hivatkozik.

42. Mit jelent a pipelining utasítás-végrehajtásban a NOP utasítások beiktatása és a data forwarding?

A memóriautasítások végrehajtásához szükséges többlet időigény és a hazardok miatti utasításvárakoztatás „üres” (NOP = No Operation) utasítások beiktatásával oldható meg.

Az „adat előreengedés” (data forwarding) azt jelenti, hogy egy utasítás eredményadata a következő utasítás számára a processzoron belül megfelelő áramkörökkel átadásra kerül, nem kell megvárni az adatnak az utasításban előírt helyre történő visszaírását.

43. Mit jelent a pipelining utasítás-végrehajtásban az utasítás-átrendezés és a scoreboarding?

A fordítóprogram (ha ez lehetséges) a program tartalmi megváltoztatása nélkül átrendezi az utasítássorrendet és a memóriautasítások és hazardok kezelése miatti várakozási időket hasznos utasításokkal tölti ki.

A "Scoreboard" egy kiegészítő vezérlőegység, amely figyeli, hogy a pipeline-ba egy "beengedni" kívánt utasításban szereplő regisztereket a pipeline-ba korábban már kibocsátott utasítások nem fogják-e módosítani.

44. Mi a célja és a megvalósítási elve a **Harvard architektúrának**?

A processzor-memória sínen az utasítás- és az adatkiolvasás (visszaírás) ütközéseit feloldhatjuk, ha az utasításokat és adatokat fizikailag különálló memóriában tároljuk, amelyek külön adat- és utasítássínen érhetők el.

45. Mit jelent a **pipeline leállítása**, illetve **törlése** a vezérlésátadó utasítások kezelésében?

1. A processzor leállítja a pipeline töltését mindaddig, míg az ugrás kimenetele nem egyértelmű.
2. A processzor mindig rögzített módon becsüli meg az elágazás kimenetelét. Ha ez nem teljesül, akkor a pipeline-ban levő utasítássort törli.

46. Mit jelent a processzor **spekulatív elágazás-feldolgozása**?

A processzor megpróbálja megjósolni – különösen a feltételes ugróutasítások esetében – a vezérlésátadó utasítás várható irányát, kimenetelét.

47. Mit jelent a vezérlésátadó utasítások **statikus** és **dinamikus** kezelése pipelining **feldolgozásnál**?

1. statikus esetben a fordítóprogram értékeli ki az ugrási feltételeket, és meghatározza a legnagyobb valószínűséggel előforduló ugrási címeket, és ennek megfelelően szervezi a pipeline-t
2. dinamikus esetben a program futása közben a processzor egy elágazástörténeti táblázatban vezeti az ugróutasítások címét és ezek kimenetét (egy vagy több biten kódolva, hogy az ugrás megtörtént-e), és ezt felhasználva jósolja meg az elágazások kimenetét

48. Milyen fontosabb problémákat kell kezelni a **szuperskalár CISC** processzorok tervezésekor, és ezekre mi a megoldás?

1. a változó hosszúságú utasítások
2. a többfajta memóriautasítás

A megoldást az x86-os utasításkészlet RISC emulációja jelentette, azaz azaz a szuperskalár CISC processzorokat egy szuperskalár RISC maggal valósították meg. Ez azt jelenti, hogy ezekben a processzorokban az x86-os utasításokat „RISC jellegű” mikroutasításokra bontják fel.

49. Miért volt szükség a szuperskalár CISC processzorokban a **regiszterek számának növelésére**, és ennek mi a következménye?

A több futószalagon egymással párhuzamosan feldolgozott, jóval nagyobb számú utasítás végrehajtása több, egy időben rendelkezésre álló regiszter meglétét is feltételezi. Ugyanakkor a regiszterek számát a kódkompatibilitás (x86-os utasításkészlet) miatt egy határon túl már nem lehetett növelni. Ezért kellett külön kezelni az utasításkészlethez tartozó architektúrát, amelyhez a programok által megcímezhető regiszterek tartoznak, és a fizikai hardverarchitektúrát, amelyhez a processzorba ténylegesen beépített regiszterek tartoznak.

50. **Hasonlítsa össze** a **skalár** és a **szuperskalár** processzorok utasítás-dekódolását és kibocsátását!

A skalár processzornak ciklusonként csak egy utasítást kell dekódolnia. Ennek során ellenőriznie kell, hogy a kibocsátandó utasítás függ-e a végrehajtás alatt állóktól. A szuperskalár processzornak ciklusonként 4 utasítást kell dekódolnia. Ellenőriznie kell a 4 kibocsátásra váró utasítás függ-e a végrehajtás alatt állóktól, és hogy a 4 kibocsátásra váró utasítás között vannak-e függőségek.

51. **Miért** van szükség szuperskalár processzoroknál **elődekódolásra**, és ennek végrehajtása **hogyan** történik?

A függőségvizsgálatok jóval nagyobb feladatot jelentenek, mint a skalár processzoroknál, ezért a processzor dekódolási feladatait az elődekódolással próbálják meg csökkenteni. Az elődekódolás során a dekódolás feladatainak egy része már akkor végrehajtásra kerül, amikor az utasításokat a másodlagos gyorsító tárból vagy a memóriából az L1 szintű gyorsítótárba írják. Ennek során ún. elődekódoló bitekkel egészítik ki az utasítást.

52. Mit jelent a szuperskalár **utasításkibocsátási ráta** és **kibocsátási politika**?

A szuperskalár utasítás-kibocsátásnak két összetevője van:

1. A kibocsátási ráta adja meg a processzor által ciklusonként kibocsátható, vagyis a végrehajtó egységekhez továbbítható utasítások számát.
2. A kibocsátási politika határozza meg a függőségek kezelését. Ezek lehetnek:
 - Adatfüggőségek
 - Vezérlésfüggőségek

53. Milyen **kezelési módjai** vannak szuperskalár processzoroknál az **adat-** és **vezérlésfüggőségeknek**?

1. Az adatfüggőségek többségét a processzor regiszter-átnevezéssel megszüntetheti (lásd következő fejezet).
2. A vezérlésfüggőségeket a processzor kezelheti:
 - blokkolással, azaz a processzor felfüggeszti az utasítás-kibocsátást a feltétel kiértékeléséig
 - spekulatív elágazáskezeléssel, azaz a processzor valamilyen módszerrel megbecsüli az elágazási utasítás kimenetét

54. Mit nevezünk **várakoztatásnak**?

Az adat- és vezérlésfüggőségek okozta blokkolódásokat a processzor kezelheti az utasításkibocsátás blokkolásával, vagy a függő utasítások pufferelemzésével a végrehajtó egység előtt, amit várakoztatásnak nevezünk.

55. Milyen **okokra** vezethetők vissza a hivatkozott **regiszterek közötti adatfüggőségek**?

1. RAW (Read After Write) típusú függőség, amikor egy utasításnak egy másik, még befejezetlen utasítás eredményadatára van szüksége.
2. WAR (Write After Read) típusú függőség, amikor egy utasításnak egy olyan regisztert kell írnia, amelyet egy megelőző utasítás még használ.
3. WAW (Write After Write) típusú függőség, amikor több utasítás azonos regisztert kíván egy időben írni.

56. Ismertesse a **regiszterátnevezés** célját, megvalósítását és lépéseit!

A regiszter-átnevezés célja az áladatfüggőségek megszüntetése.

A regiszter-átnevezés megvalósítása lehet:

1. statikus (fordítóprogram hajtja végre)
2. dinamikus, amikor az átnevezést futás közben a hardver hajtja végre.

A regiszter-átnevezés során a processzor egy leképező-táblázattal hozzárendeli az ISA architektúrais regisztertárat az átnevező regisztertárhoz.

Az átnevezés folyamata a következő lépésekkel valósul meg:

1. az utasítás-kibocsátáskor az utasításban hivatkozott cél- és forrásregiszter átnevezése az átnevező regisztertárral
2. a kiküldéskor a regiszter érvényességi bitek, azaz a függőségek ellenőrzése
3. a végrehajtás után az átnevező regisztertár aktualizálása

57. **Spekulatív elágazás-feldolgozás** esetén milyen **módszerei** vannak az elágazás becslésének?

A feltételes elágazási utasítások lehetséges kimeneteinek becslése történhet:

1. egykimenetűen, amikor a becslés rögzítetten mindig azonos (elágazik vagy nem)
2. kétkimenetűen, amelynél statikus esetben a fordítóprogram a tárgy kód elemzése alapján, vagy dinamikus esetben a program futása során a processzor az egyes elágazások megvalósulásának statisztikája alapján „jósolja” meg az elágazást

58. Mit jelent a **soros konzisztencia megőrzése** szuperskalár processzoroknál?

1. a programutasítások soros végrehajtásának helyreállítása (processzor-konzisztencia)
2. a memória-hozzáféréseknek az eredeti utasítássorrend szerinti végrehajtása (memória konzisztencia)

59. Milyen **állapotban** lehetnek a **programutasítások szuperskalár** feldolgozásnál?

1. dekódolt
2. kibocsátott (várakozási pufferbe beírt)
3. végrehajtás alatt
4. végetért
5. befejezett

60. Mi a **ROB** és segítségével hogyan dönthető el, hogy egy utasítás befejeződött-e?

A soros konzisztencia biztosításának legfontosabb eszköze a szuperskalár processzoroknál az átrendező puffertár, a ROB (ROB = Re Order Buffer), amelyben az eredeti utasítássorrendben (in-order) lefoglalásra kerül egy hely az utasításnak a várakoztató pufferbe történő írásával egy időben.

Az utasításról el tudjuk dönteni hogy:

1. végetért, azaz a megfelelő végrehajtó egység befejezte az utasítást és az eredményadat egy átmeneti tárolóba (a regiszter-átnevezéskor kijelölt fizikai regiszter) került, illetve
2. befejezett, ha az eredményt a processzor beírta az utasításban előírt végleges tárolóhelyre (ISA regiszter vagy memória)

61. Mit jelent az **IA-64 bites EPIC** architektúra? Hol alkalmazzák az ilyen processzort?

Az IA-64 architektúra nem egyszerűen az IA-32-es processzor-architektúra 64 bites címezéssel működő változata, hanem a 64 bites RISC processzorok felépítését és működési elveit adaptálja illetve fejleszti tovább. Az EPIC = Explicitly Parallel Instruction Computing. Többprocesszoros szerverekben, munkaállomásokban használják.

62. Mi a szerepe a fordítóprogramnak az IA-64-es ISA architektúrában?

A fordítás során a programozási nyelv fordítóprogramjának nem soros, hanem eleve párhuzamosított tárgykódot kell generálnia a processzor számára.

63. Hogyan történik az elágazások statikus kezelése az IA-64-es architektúrájú processzoroknál?

Az IA-64-es architektúra az elágazások előrejelzését a RISC processzorokhoz hasonlóan első szinten a fordítóprogramra bízta.

64. Mit jelent az elágazások kezelésében a predikációs technika az IA-64-es architektúrában?

A predikációs technika lehetővé teszi, hogy egy feltételes elágazásnál a processzor mindkét programágat egyszerre, párhuzamosan hajtsa végre. Az elágazás kiértékelése után szükségtelen ág adatait egyszerűen törli a processzor.

65. Mit jelent a cikluszigöngyölítés és a spekulatív betöltés a memóriából?

Az architektúra fordítóprogramja a legtöbb esetben meg tudja állapítani, hogy egy adott cikluson belüli kódblokk hányszor kerül végrehajtásra. Így alkalmazható a RISC processzoroknál használt megoldás, a ciklus úgynevezett "kigöngyölítése". Ez azt jelenti, hogy a cikluson belüli kódblokkot annyiszor ismétli meg a fordítóprogram, amennyiszer azt végre kell hajtani. Az IA-64 architektúra egy spekulatív technikát alkalmaz elágazásoknál: a memóriából történő, előre ütemezhető betöltéseket a processzor végrehajtja mindkét programágon.

2. ZH

Tárolókezelés

1. Definiálja az elérési időt és a ciklusidőt!

Elérési idő: A memória megcímezése után az adatnak a tároló kimenetén való megjelenéséig eltelt idő.

Ciklus idő: Elérési idő + feléledési idő, azaz az az idő, amelynek elteltével egy címzés után a memória újra megcímezhető.

2. Jellemezze a tárolóhierarchiát a hozzáférés gyakorisága, az idő, a kapacitás és az ár szerint!

A számítógépben lévő adattároló-egységek felépítése hierarchikus, amelyet az jellemez, hogy az adathozzáférések gyakorisága szerint a processzor közelében igen gyors elérésű, kis kapacitású és drága tárolók helyezkednek el, majd ahogy távolodunk a processzortól az adattárolók elérése lassul, kapacitásuk nő és az egységni információátvitel költsége is csökken.

3. Mit nevezünk regiszternek?

A processzor tartalmazza a regisztereket, amelyek egyenként általában 64, 32, 16 bit információ átmeneti tárolását végzik. Ezek a számítógép leggyorsabb, ugyanakkor legdrágább tárolóegységei.

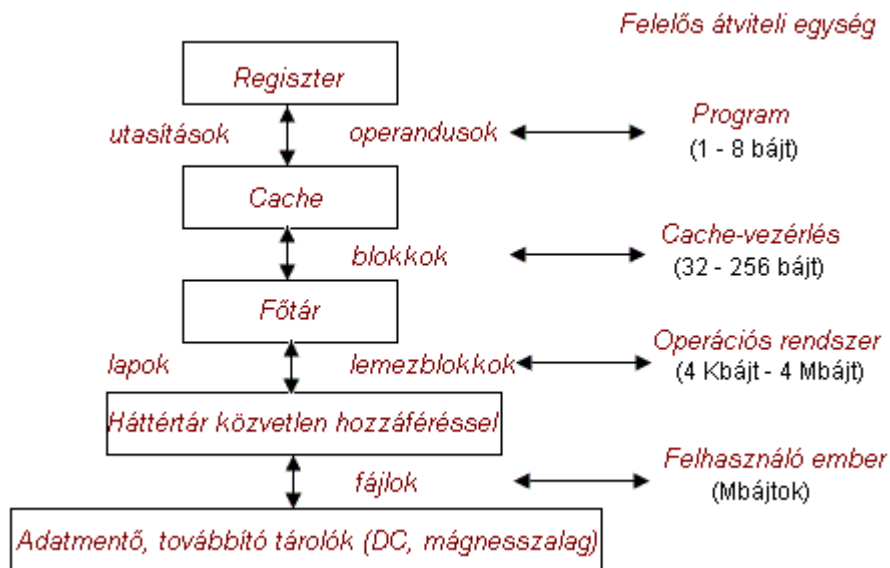
4. Mekkora a hozzáférési ideje a **cache** tárnak, a főtárnak és a háttértárnak a regiszterekéhez képest?

A gyorsító (cache) tárok tárolókapacitása - amelyek a futó programból a processzor számára leggyakrabban szükséges utasításokat és adatokat tartalmazzák - már nagyságrendekkel nagyobb a regiszterekénél, ugyanakkor az adatok kiolvasása kb. 2-szer annyi időt igényel, mint a regisztertáraknál.

5. Mi a **tárolókezelés** alapfeladata?

A tárolókezelés alapfeladatát a következőképpen fogalmazhatjuk meg: biztosítsa a processzor műveletvégrehajtásához szükséges adatokat, összehangolva a tárolóhierarchia egyes szintjein lévő memóriaegységek működését.

6. Adja meg, hogy a **tárolóhierarchia** szintjei között milyen formában **áramlanak** az **adatok** és az adatmozgatásért felelős egységet!



7. Jellemezze a **regiszterbankot**, az **ablak-** és a **blokktechnikát**!

A regiszterbank (register banking) esetében a regisztertömb fix méretű (azaz meghatározott darabszámú regiszterből álló) részekre van felosztva, melyek a felhasználói taszkokhoz kerülnek hozzárendelésre.

Az ablaktechnika (register windowing) esetében a regisztertömb szintén fix méretű részekre van felosztva, de ezeknek lehetnek az egyes taszkok között átlapolható részei (azaz ez esetben egy taszk egy másik taszk regisztereinek egy részét is „láthatja”).

A blokktechnikánál (register blocking) a regisztertömb változó méretű átlapolható részekre van felosztva.

8. Mit nevezünk **cache** tárolónak? Miért „intelligens” a cache vezérlő?

A cache vagy más néven gyorsító tárolók az utasítások és adatok átmeneti tárolására szolgáló, viszonylag kisméretű, gyors memóriapufferek, amelyek a főtár blokkjainak másolatát tartalmazzák, önálló vezérléssel rendelkeznek és a felhasználói programok számára láthatatlanok (elérhetetlenek). A cache fogalmának meghatározásában blokk alatt a főtár rekeszeinek olyan egymás utáni sorozatát értjük, amelynek bemásolása a cache-be egy lépésben megtörténhet. A cachevezérlő egy „intelligens” áramkörökből felépített hardveregység.

Intelligencia alatt ez esetben az értendő, hogy például a vezérlő képes a különböző bemásolási és adatviisszaírási stratégiák, algoritmusok kezelésére.

9. Mit jelent az **L1** és **L2** cache?

Mikroszámítógépeknél a CPU és a központi memória között található cache tárat általában több szinten valósítják meg. Így a processzortól kiindulva megkülönböztetünk L1, azaz elsőszintű (Level 1) cache tárolót és L2 másodszintű (Level 2) cache-t.

10. Mit jelent a programok **időbeli és helyi lokalitásának** elve?

Időbeli lokalitás: ha egy adatra vagy egy utasításra hivatkozás történik, akkor ez nagy valószínűséggel rövid időn belül ez újra megtörténik. (Ciklus)

Helyi lokalitás: ha egy adatra vagy egy utasításra hivatkozás történik, akkor ez nagy valószínűséggel a környezetében lévő címekre is megtörténik.

11. Mi a **cache működési elve**?

Ha a processzor a központi tár egy rekeszét olvasni akarja, akkor a gyorsító tár vezérlője bemásolja ezt és még néhány rekeszt a cache tárolóba. Ugyanis a lokalitás elve alapján nagy valószínűséggel feltételezhető, hogy a processzornak legközelebb a soron következő rekeszre lesz szüksége. Ha ez így van, a processzor a következő olvasási műveleteket már várakozás nélkül képes lesz végrehajtani, mindaddig, amíg a szükséges adatok a gyorsítótárban rendelkezésre állnak.

12. Melyek a **cache** tároló legfontosabb **jellemzői**?

1. a cache-tár mérete, a blokk mérete (az adatcsere a főtár és a cache között mindig blokkos formában történik)
2. egy blokk kikeresésének és azonosításának eljárása a cache tárban
3. aktualizálási eljárás, amely szerint a processzor által a cache-tárban módosított adatot a főtárba írjuk
4. a megfelelő helyettesítési stratégia (replacement policy), amivel eldöntjük, hogy a cache-ben melyik blokkot lehet felülírni, ha új blokkot kell bemásolni
5. a főtár és a cache-tár adategyezőségének biztosítása

13. Mit nevezünk **cache hit-nek** és **cache miss-nek**?

Ha a processzor olyan adatot igényel, amely a cache-ben megtalálható, akkor találatról vagy cache hit-ről beszélünk. A találatot a cache-vezérlő azzal állapítja meg, hogy a bemásolt blokkokhoz tartozó címrészek (toldalék vagy „tag”) alapján valamelyik cache blokkban benne van-e a processzor által igényelt adat főtárbelicíme.

Ha a processzor által igényelt adat nincs meg a cache-ben, ezt tévesztésnek vagy cache miss-nek nevezzük.

14. Adja meg, hogy a **cache átlagos hozzáférési idejét** hogyan lehet **kiszámítani**!

$$(HIT - HF) \frac{[100 - (MISS - RATE)]}{100} + \frac{MISS - RATE}{100} \cdot (MISS - JAV)$$

15. Mit tartalmaz a cache **toldalék** (tag) és **adatrésze**?

A toldalek (tag) egyrészt a főtárból bemásolt blokkra vonatkozó címinformációkat tartalmazza (címrész), másrészt itt kerülnek bitenként kódolva letárolásra a cache blokk adataira vonatkozó érvényességi információk (vezérlő rész).

Az adatrész tartalmazza a bemásolt főtárblokk változatlan vagy a processzor által már módosított adatait.

16. Definiálja az **asszociatív tárolókat**! Mi szerint keresünk vissza az asszociatív tárolóként működő cache-ben?

Asszociatívnak nevezzük azokat a tárolókat (CAM = Content Addressable Memory), amelyek az adatok visszakeresését tartalom szerint végzik, azaz a tárolóban lévő adatokat nem kell megcímezni, hanem a keresett adatokat a tároló bemenetére helyezve eldönthető, hogy az adatokat a tároló tartalmazza-e vagy sem.

Az asszociatív tárolóként működő cache-ek esetében az adatok, amelyek tartalma szerint keresünk vissza, a főtárbeli blokkok címei vagy részcímei.

17. Jellemezze a **teljesen asszociatív cache** tárolót!

A teljesen asszociatív cache tárolóknál egy főtárbeli blokk a cache bármelyik sorába bemásolható, a blokk címe (sorszám) pedig bekerül a cache toldalek részébe. Ha a processzor egy adatot keres a cache-ben, akkor az adat memóriacíméből képzett blokkorszám asszociatív módon összehasonlításra kerül a cache-ben lévő blokkok sorszámaival. Az összehasonlítás rendkívül gyorsan, minden sorra vonatkoztatva azonos időben történik meg. Ehhez annyi összehasonlító áramkörre van szükség, amennyi sort tartalmaz a cache. Emiatt ezt a cache típust nagyon gyorsnak és rugalmasnak, egyúttal nagyon drágának jellemezhetjük.

18. Jellemezze a **direkt** (közvetlen) **leképezésű cache** tárolót!

A közvetlen leképezésű cache tárolóban egy blokk csak a cache egy konkrét sorába kerülhet. Ez a sor úgy kerül meghatározásra, hogy a blokkorszám alsó „n” db bitjét használjuk fel sorindexnek. A közvetlen leképezésű cache-k esetében ezért a különböző blokkokban lévő, de azonos sorindexű blokkokban található adatok hozzáférése rendkívül lelassul, mivel ez minden esetben blokkcserét eredményez a cache-ben. Ez azt jelenti összességében, hogy a találati arány is kisebb lesz az asszociatív tárat használó cache-nél. Ugyanakkor azt is meg kell jegyezni, hogy a közvetlen leképezésű cache tárolók olcsók és gyors visszakeresést biztosítanak.

19. Jellemezze az **n-utas csoport asszociatív cache** tárolót!

Az „n”-utas csoport asszociatív gyorsító tárolóknál a cache több, „n” sorból álló részre, csoportokra van osztva. Az egy csoporthoz tartozó cache tárolórész önmagában teljesen asszociatív tárolóként működik, azaz egy cache-be másolandó blokk a csoporton belül az „n” sor bármelyikébe bekerülhet. Annak megállapítása, hogy egy cache-be írandó blokk melyik csoporthoz tartozik, a közvetlen leképezésű cache tárolóknál megismert módon, a memóriacíméből képzett index-szel kerül meghatározásra. A csoport asszociatív cache tárolók rugalmasabbak, mint a közvetlen leképezésűek és relatíve kevesebb asszociatív összehasonlító áramkört tartalmaznak, és még viszonylag gyorsak. Ezért a gyakorlatban ez a cache típus terjedt el szélesebb körben.

20. Mit értünk **helyettesítési stratégia** alatt? Melyik a legelterjedtebben alkalmazott formája?

Ha egy főtárblokkot be kell másolni a cache-be, az első kérdés annak eldöntése, hogy a bemásolandó blokk adatai a cache melyik sorába kerülhetnek be, azaz az új blokk adataival, melyik más régebben a cache másolt blokk adatait lehet felülírni. Erre a célra a cache vezérlésében különböző helyettesítési stratégiákat alkalmaznak. Ezek közül a legelterjedtebb az LRU (least recently used) stratégia, amely esetében a processzor által legrégebben használt blokk adatai kerülnek felülírásra.

21. Mit jelent a **demand fetching** és a **prefetching**?

A demand fetching azt jelenti, hogy csak a processzor adatigénye esetén keresik ki a főtárból a megfelelő blokkot és töltik be a cache-be. Ezzel párhuzamosan a processzor is azonnal megkapja az adatot (load through).

A prefetching azt jelenti, hogy ha a főtárból be kell tölteni egy blokkot a cache-be, akkor automatikusan betöltésre kerül a főtár következő blokkja is.

22. Jellemezze a **közvetlen átírást (write through)**! **Hogy javítható** a módszer hatékonysága?

A közvetlen átírás (write through) eljárásnál a gyorsítótár írásával együtt megtörténik a főtár írása is. Ennek alkalmazása esetén a főtár és a cache adategyezősége "automatikusan" biztosított, de a főtár írásműveleteinek végrehajtási idejét a cache alkalmazása nem javítja.

A közvetlen átírás módszerének hatékonyságát javítja a pufferezt közvetlen átírás (buffered write through), amely esetben a processzor a megváltoztatandó főtárbeli adatokat egy (tipikusan 4 elemű) íráspufferbe írja be és nem várja meg a főtár írásának a befejeződését.

23. Jellemezze a **visszaírást (write back)**!

A visszaírás (write back) eljárásnál a gyorsítótárban módosított adat csak akkor kerül visszamásolásra a főtárba, ha a cache-nek a módosított adatot tartalmazó sorát felül kell írni egy, a főtárból bemásolandó újabb blokkal. Ez a módszer gyorsabb a közvetlen átírásnál, viszont minden cache-beli sor esetében meg kell jegyezni, hogy az adott sor módosításra került-e. E célra soronként egy kiegészítő bit szolgál, amelynek a "módosult" (alter) vagy "piszkos" (dirty) nevet adták. Ha írás történik a cache-be, akkor megfelelő sorban a "módosult" bit beállításra kerül. Ha a sor tartalmát cserélni kell egy másik főtár blokkal, akkor a módosult bit beállított állapota jelzi a cache vezérlőnek, hogy a sor tartalmát előzőleg vissza kell írni a főtárba.

24. Mi a **MESI**, és milyen állapotban lehetnek alkalmazása esetén a cache blokkok?

A MESI egy szabványeljárás, amely a főtár és a cache táruk azonosságát biztosítja. Lehetséges állapotai:

1. módosított
2. kizárólagos
3. megosztott
4. érvénytelen

25. Jellemezze a **hardver-lemezgyorsító tárak** működését!

Ha a processzor adatokat akar kiírni a lemezre, akkor a gyorsító vezérlője ezeket letárolja a cache-ben és visszaigazolja a processzornak a lemezművelet végrehajtását. Ezt követően kezdi meg a vezérlő az adatoknak a kiírását a lemezre, és így ennek ideje alatt a processzor folytathatja a műveletvégrehajtást.

Ha a processzor adatokat akar olvasni a lemeztől, a gyorsító vezérlője megvizsgálja, hogy az adatok megtalálhatóak-e a cache-ben. Ha igen, akkor az adatokat közvetlenül a cache-ből továbbítja, lemezművelet végrehajtása nélkül.

26. Jellemezze a **ROM tárolókat (ROM, PROM, EPROM, EEPROM)**!

A ROM (Read Only Memory = csak olvasható memória) gyárilag beprogramozott, csak olvasható és a felhasználó által megváltoztathatatlan memória. Tartalma a számítógép kikapcsolása után is megőrződik.

PROM = Programable (programozható) ROM. Ebbe a memóriatípusba a felhasználó egyetlen alkalommal beírhatja a számára állandóan megőrzendő adatokat vagy programokat. Ezt követően a PROM tartalma már nem változtatható;

EPROM = Erasable Programmable ROM tároló az adatokat elektromos töltés formájában őrzi meg. Ez a ROM tároló ibolyántúli (UV) fénnel törölhető, majd újra írható. Ehhez természetesen megfelelő készülék és szakértelem szükséges.

EEPROM = Electrically Erasable Programmable ROM tárolók újraprogramozásuk előtt egy elektronikus impulzussal törölhetők.

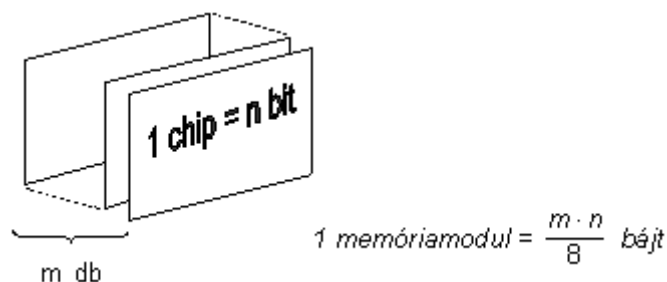
27. Mit jelent az **árnyék-** és a **flash-memória**?

A ROM tárolók elérési ideje kialakulásuk idejében a RAM tárolókéval körülbelül azonos volt, ma már viszont lényegesen lassúbbak. Ezért a gyakran használatos programrészeket (pl. PC-knél a BIOS-t) a mai gépekben átmásolják egy RAM területre. Ezt árnyék memóriának (shadow) nevezzük.

A flash memóriák alapját az EPROM technológia szolgáltatja, de működésükben az a különbség, hogy blokkonként törölhetők és elektronikusan újraírhatók.

28. Mi **jellemzi** a **RAM** chipeket? Hogy számítjuk ki egy RAM memóriamodul **kapacitását**? Mit jelent a **SIMM** és **DIMM** memóriamodul?

A RAM (Random Access Memory = véletlen elérésű memória) a programok által írható és olvasható memória. Tartalma a számítógép kikapcsolása után visszavonhatatlanul elvész. Kapacitásának kiszámítási módja:



A SIMM (single in-line memory module) és DIMM (dual in-line memory module) egy sorozat RAM integrált áramkört tartalmaz. A SIMM memóriamoduloknak a két oldalán redundáns érintkezők vannak, a DIMM két oldalán különböző érintkezők vannak, illetve a SIMM-nek 32 bites adatcsatornája van, a DIMM-nek pedig 64 bites.

29. A memóriáknál milyen **hibakezelési eljárások** vannak?

A memóriahibák felismerését a paritásellenőrzés szolgálja. Ehhez minden bájt kiegészítésre kerül egy paritásbittel, ami alapján a memóriavezérlő a hibás memóriaműveleteket felismeri. Ha olyan memóriát kívánunk alkalmazni, amely nem csak felismeri, hanem ki is javítja például az 1 bites paritáshibákat, akkor ehhez ún. ECC = Error Correction Circuit hibajavító áramkörökre van szükség.

30. Definiálja az **SRAM-ot** és a **DRAM-ot**!

A statikus RAM-ok (SRAM) az adatokat félvezető memóriában (flip-flop) tárolják. Ennek állapota mindaddig fennmarad, amíg ezt újabb írással meg nem változtatjuk, vagy a tápfeszültség meg nem szűnik. Ez azt jelenti, hogy az SRAM-okat nem kell időközönként frissíteni, ciklusidejük lényegében megegyezik az elérési idővel.

A dinamikus RAM-ok (DRAM) elemi cellája egy igen kis méretű kondenzátor, néhány pF (pikoFarad) kapacitással. Ezek az elemek viszonylag nagy sűrűséggel helyezhetők el az integrált áramkörben, előállításuk relatíve olcsó. Az elemi cellák (kondenzátorok) töltése viszont egy idő után kisül, ezért ennél a memóriatípusnál a memória tartalmát meghatározott időközönként fel kell frissíteni.

31. Ismertesse a **DRAM felépítését**! Hogyan címezünk egy memóriacellát a memóriamátrixban?

A DRAM mátrixszervezésű azaz oszlopokból és sorokból épül fel, amelyek „metszéspontjában” található egy memóriacella. A memóriacellát megcímezni két lépésben lehet:

1. először a mátrix egy sorát címezzük meg, ezt követően
2. a sorból kiválasztjuk az adott oszlophoz tartozó cellát

32. Mi célt szolgál a **RAS** és **CAS** vezérlőjel? És a **WE**?

A multiplexált sor- és oszlopcímzés két vezérlőjelet:

1. RAS = Row Address Strobe (sorcím-kiválasztó impulzus)
2. CAS = Column Address Strobe (oszlopcím-kiválasztó impulzus)

használ, amelyek jelzik a DRAM-csipnek, hogy a megfelelő sor- ill. oszlopcím a címpufferbe elhelyezésre került. A DRAM olvasását ill. írását a WE = Write Enable vezérlővonal alacsony ill. magas jelszintje különbözteti meg.

33. Hogy épül fel a **DRAM** egy **cellája**? Mit értünk **frissítés** alatt?

A DRAM IC-kben egy bit tárolását biztosító elemi cella egy kondenzátorból és egy tranzisztorból épül fel. Mivel valójában a tranzisztor zárt állapotában is a kondenzátor folyamatosan veszít a töltéséből, ezt periodikusan újra kell tölteni, azaz frissíteni (refresh).

34. Mit jelent, hogy egy **16 Mbites DRAM-chip 2Mx8** szervezésű? Milyen leképzésű ez a chip?

Itt az első szám az egyes tárolómezők kapacitását adja meg Mbit-ben, a második szám pedig a párhuzamos működésre képes tárolómezők darabszámát jelenti. 2Mx8-as szervezésű 16 Mbites DRAM-csipnek 12 cím- és 8 adatvezetékekkel kell rendelkeznie, ha a sor- és oszlopcímzés multiplexált.

35. Milyen lehetőségeink vannak **DIMM-modul előállítására** 16 Mbit-es DRAM csipekből?

Ha 16 Mbit-es DRAM csipekből 64 bites adatátvitelre képes DIMM (Dual In-Line Memory Module) memóriamodulokat állítunk össze, a következő lehetőségek vannak:

- 16 db 4Mx4-es szervezésű chip = 32 Mbájtos memóriamodulok
- 8 db 2Mx8-es szervezésű chip = 16 Mbájtos memóriamodulok
- 4 db 1Mx16-es szervezésű chip = 8 Mbájtos memóriamodulok

36. Mit jelent a RAS előtöltési idő (**t_{RP}**)? És a ciklusidő (**t_{RC}**)?

TRP = RAS előtöltési idő (RAS - Precharge Time):

Az az időtartam, amely alatt a memória áramkörei visszaállnak az újabb hozzáféréshez szükséges alapértékekre. Ezt feléledési időnek is szokták nevezni, értéke megegyezik két RAS jel közötti időtartammal.

TRC = Ciklusidő (Cycle Time):

Két memóriahozzáférés közötti idő.

37. Definiálja a **RAS** és **CAS hozzáférési időt**!

TRAC = RAS hozzáférési idő (RAS Access Time):

A RAS vezérlőjel kiadása és az adatvezetéken az adat megjelenése között eltelt idő.

TCAC = CAS hozzáférési idő (CAS Access Time):

A CAS vezérlőjel kiadása és az adatvezetéken az adat megjelenése között eltelt idő.

38. Mit jelent a **RAS-CAS késleltetési idő**?

TRCD = RAS-CAS késleltetési idő (RAS-CAS-Delay):

Az az időtartam, amely után a RAS vezérlőjelet a CAS vezérlőjel követi. Értékét meghatározza, hogy a sorcím kiadása után mennyi idő múlva képes a memória az oszlopcímet fogadni.

39. Ismertesse a **memory interleaving** eljárás lényegét!

Gyorsítás átlapolt memóriakezeléssel (Memory interleaving):

a memóriát egymástól függetlenül címezhető és olvasható részekre, úgynevezett memóriabankokra osztjuk fel. Feltételezve, hogy a processzor általában a memóriát címfolytanosan olvassa, a 0. memóriabankból kiolvasott adat hozzáférése alatt az 1. memóriabankban lévő következő címen lévő adat már megcímezhető.

40. Ismertesse a **programlokalitás** és **-gyakoriság elvét** és összefüggését a virtuális tárkezeléssel!

Programlokalitás elve:

A programok végrehajtásuk során legtöbbször egy korlátos memóriaterületen belül dolgoznak, igen ritka például az olyan ugró utasítás, mely a program elejéről a végére adja át a vezérlést.

Gyakoriság elve:

Az egyes programrészek között igen nagy eltérések lehetnek abból a szempontból, hogy a programvégrehajtás során milyen gyakran van rájuk szükség.

41. Mit nevezünk **virtuális tárnak és virtuális címnek**?

Jelöljük ki a háttértárolón a főtár memóriakapacitásánál jóval nagyobb tárolóterületet úgy, hogy az egy időben aktív programfolyamatokhoz tartozó programok és adatállományok ezen elférjenek. Nevezzük ezt látszólagos, azaz virtuális tárterületnek. A háttértárolón kijelölt virtuális tárolót osszuk fel blokkokra, amelynek méretét a lokalitás elvének figyelembevételével határozzuk meg.

A virtuális tárolóban 1 bájtot virtuális címekkel lehet megcímezni, mely két alapvető részből áll: egyrészt megadjuk, hogy a keresett bájt a virtuális tár hányadik blokkjában található, továbbá azt, hogy az adott blokk kezdőcímétől számítva hányadik bájt (relatív cím).

42. A virtuális tárkezelésnél hogyan választjuk meg a **blokk méretét**? Hova lehet egy **blokkot beírni a főtárba**?

Viszonylag nagy tárterületet célszerű blokknak kijelölni, annak érdekében, hogy gyakran ne legyen szükség a blokk cseréjére, mivel ehhez relatíve hosszú beolvasási idő szükséges. Egy blokkot tetszőleges helyre lehet beírni a főtárba.

43. A virtuális tárkezelésnél **blokkcsere** esetén melyik főtár blokkot írjuk felül? A megváltoztatott blokkot mikor írjuk vissza a virtuális tárba?

Azt, hogy melyik főtár blokkot írjuk felül az operációs rendszer dönti el algoritmussal, leggyakoribb az LRU (Least Recently Used) stratégia alkalmazása, amely a processzor (programok) által legrégebben használt blokk kiválasztását eredményezi.

Ha megváltozott blokkot szeretnénk visszairni a virtuális tárba, a write back eljárás alkalmazása a célszerű, azaz csak blokkcserénél aktualizáljuk a virtuális tárat, mert az aktualizálás jelentős időt igényel.

44. Milyen címeket tartalmaznak **virtuális tárkezelésnél a programok**? Hogyan **képezzük le a virtuális címet a főtár fizikai címére**?

A számítógépen aktuálisan futó programfolyamatok utasításai a virtuális címeket, mint logikai címeket tartalmazzák. A virtuális tár blokkjai akkor kerülnek bemásolásra a főtárba, ha valamilyen programutasításban hivatkozás történik az adott blokkban található címre és az nem található meg még a központi memóriában. Az előbbiekből következik, hogy a programok a virtuális tárat úgy látják, mintha az a központi tár lenne. A virtuális címmel elvileg megcímezhető memóriaterületet virtuális címtartománynak nevezzük. Természetesen a processzornak a műveletvégrehajtás során a főtár valódi, vagy másképpen nevezve fizikai címekre van szükség, tehát a virtuális címeket át kell alakítani fizikai címekké. Ehhez két dolog szükséges:

1. a blokkok háttértárolón található címét, a fizikai memóriába bemásolt blokkok sorszámát, a fizikai kezdőcímét stb. megfelelő táblázatokban nyilván kell tartani
2. egy olyan program vagy hardveregység, amely a memóriába bekerült blokkok adatait tartalmazó táblázatok alapján elvégzi a virtuális cím fizikai címmé történő átalakítását

45. Fejtse ki a **MMU** feladatait virtuális tárkezelésnél!

Az MMU a címleképezés mellett figyeli, hogy a végrehajtandó utasításban olyan virtuális cím van-e, amely olyan blokkra hivatkozik, amely még nem került be a főtárba. Amennyiben egy utasításban hivatkozott blokk nincs bemásolva a főtárba, akkor ez kivételt okoz, és az operációs rendszer megszakításkezelő rutinja meghatározza azt a főtárba már bemásolt blokkot, amelyre várhatóan legkevésbé

lesz szükség az elkövetkezendőkben, és ez kiírásra kerül a háttértáron lévő virtuális tárba, helyére pedig bemásolásra kerül az a blokk, amelynek adataira az aktuális utasítás végrehajtásához szükség van.

46. Mit nevezünk **szegmensnek**? Szegmentálásnál hogyan képezzük le a virtuális címet fizikai címmé?

Ha a virtuális tár olyan logikai blokkokból áll, amelyeknek mérete nem rögzített, akkor ezeket a blokkokat szegmenseknek, a virtuális tárkezelésnek ezt a formáját pedig szegmentálásnak nevezzük. A szegmentálásnál a logikai cím a szegmens sorszámát és a megcímzett bajtnak a szegmenskezdetétől való relatív címét tartalmazza. A szegmens fizikai kezdőcímét a szegmenstáblázat tartalmazza, ebből a szegmens sorszáma – amelyet szelektornak is neveznek – alapján lehet kikeresni a konkrét szegmens főtárbeli kezdőcímét.

Ezt követően a **fizikai cím = szegmens fizikai kezdőcíme + relatív cím** összefüggéssel meghatározható.

47. Milyen előnyei és hátrányai vannak a szegmentálásnak?

- a szegmentált virtuális tárkezelés előnye a rugalmasság (a változtatható blokkméretek miatt), az osztott felhasználás lehetősége az átlapolódó szegmensekkel
- hátránya, hogy a nagyméretű szegmensek cseréje ronthatja a hatékonyságot, és a memória teljes körű kihasználtsága sem biztosított.

48. Mit nevezünk **lapnak** és **lapkeretnek**? Lapozás esetén mit tartalmaz a **virtuális cím**?

Ha a virtuális tár rögzített méretű nem átlapolható blokkokból áll, akkor ezeket lapoknak nevezzük, a virtuális tárkezelésnek ezt a formáját pedig lapozásnak. A főtár a lapmérettel megegyező nagyságú részekre van felosztva, ezeket lapkereteknek (frame) nevezik. A virtuális cím a lap sorszámát és a megcímzett bajtnak a lap kezdetétől számított relatív címét tartalmazza.

49. Mit tartalmaznak a **laptáblázatok**?

A központi memóriába beolvasott lap fizikai kezdőcímét a főtárban – azaz annak a lapkeretnek a címét, ahová a lap elhelyezésre került – a laptáblázatok tartalmazzák. Általában szintén a laptáblázat tárolja a főtárban nem szereplő lapok lemezcímét. A laptáblázatban egy érvényességi (valid) bit jelzi, hogy a megcímzett lap betöltésre került vagy sem a főtárba.

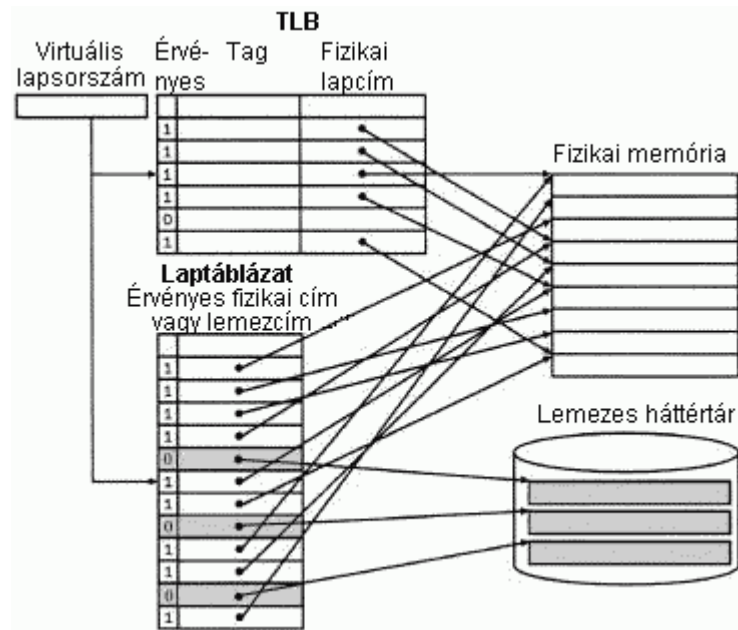
50. Hogyan történik a **címleképezés** lapozásnál, hogyan kezelhető a **laphiba** kivétel?

Minden programfolyamatnak (processznek) saját laptáblázata van, ennek kezdőcíme sokszor egy regiszterben található.

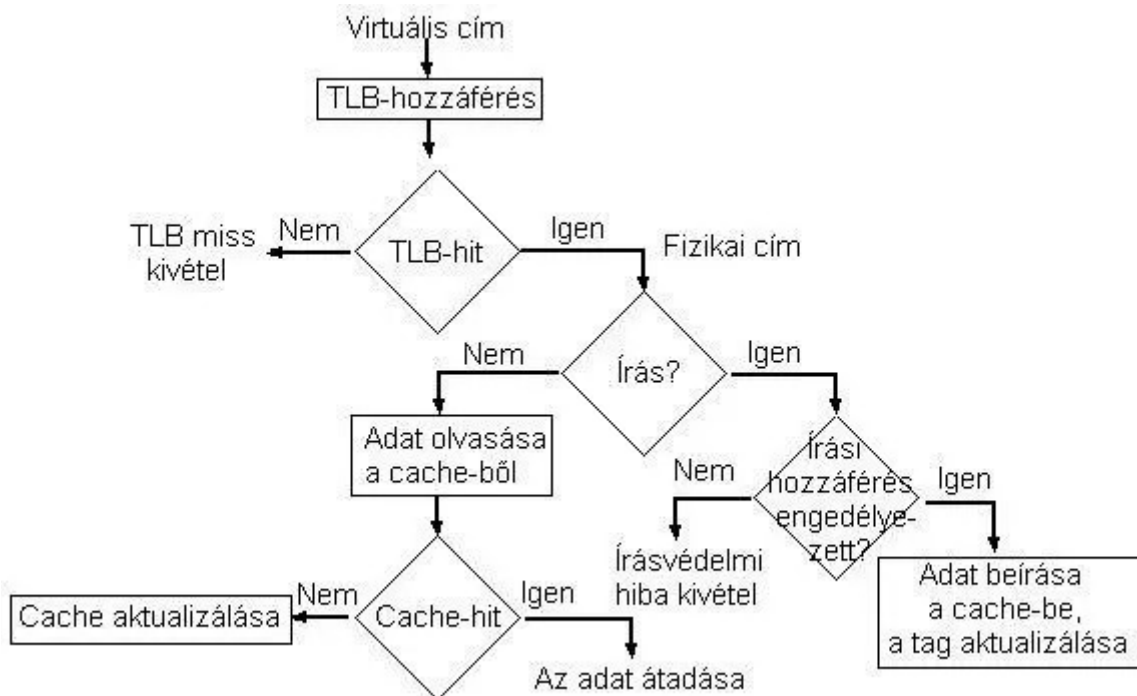
Ha egy programfolyamat egy utasítása olyan virtuális címre hivatkozik, amelynek megfelelő lap nincs a főtárban, akkor ez „laphiba” kivételt okoz. (A laphibát az MMU azáltal ismerni fel, hogy a laptáblázatban a lap érvényességi bitje 0 értékű.) Ez egy megszakítást eredményez, és a vezérlést megkapja az operációs rendszer laphiba kezelő rutinja, amely a hivatkozott lapot betölti a főtárba.

51. Mi a **TLB**? Vázolja fel a TLB-t és cache tárolót tartalmazó architektúra logikai bloksémáját!

A TLB a leggyakrabban használt lapok lapcímfordításhoz szükséges adatait tartalmazza. Fully associative, set associative vagy direct mapped cache tároló, legtöbbször 32-256 bejegyzéssel.



52. Mutassa be a **tárolóhozzáférés** folyamatát TLB-hit esetén cache tárolót is tartalmazó architektúra esetén!



53. Ismertesse a **TLB-miss** kezelésének folyamatát cache tárolót is tartalmazó architektúra esetén!

A TLB-miss esetében a laptáblázat érvényességi bitjét (Valid Bit) kell kiolvasni és az alapján:

Ha a lap a főtárban van, de nincs bejegyzés a TLB-ben (ez a „tisztá” TLB-miss), akkor a megfelelő bejegyzést a laptáblából be kell írni a TLB-be.

Ha a lap nincs a főtárban, akkor laphiba keletkezett. Ennek kezelése:

- lap címének meghatározása a háttértárban
- egy „D” = dirty lap kiválasztása (pl. LRU szerint) a laptáblában, és visszaírása a háttértárba
- a hiányzó lap bemásolása a háttértárból a főtárba
- a laptábla aktualizálása
- a TLB aktualizálása

54. Mi a lényege a **szegmentáláson belül alkalmazott lapozásos virtuális tárkezelésnek**? Milyen **címzést** használnak az **Intel** processzorok védett üzemmódban?

Lapozásos virtuális tárkezelésnél fregmentáció, azaz üres kihasználhatatlan memóriaterületek nem képződhetnek, mivel a lapok csak azonos méretű lapkeretek helyére kerülhetnek be a főtárba. Ebben az esetben a szegmentáláson belül alkalmazzák a lapozásos virtuális tárkezelést, azaz a virtuális tár szegmensei lapokból épülnek fel. Védett üzemmódban az i80386, i80486-os, valamint a Pentium processzorok, amelyek szegmenscímmel belüli kétszintű lapcímezést valósítanak meg.

55. Milyen **feladatokat** kell megoldani a **tárolóvédelmi** rendszernek?

1. a hibás vagy nem létező címek kiszűrése
2. az operációs rendszer védelme a felhasználói programoktól
3. a felhasználói programok egymástól történő védelme
4. az adatokhoz, programokhoz történő hozzáférés jogosultságainak ellenőrzése

56. Mutassa be a **privilegizálási szintekkel** megvalósított tárolóvédelem lényegét!

Minden egyes taszkhoz hozzárendelésre kerül egy privilegizálási szintszám, amelyhez meghatározott jogosultságok, illetve tiltások kapcsolódnak. Legyen magas jogosultsággal rendelkező 0-ás privilegiumszinten futó taszk 1 és taszk 2. Ezek saját szintjükön egymás rutinjait meghívhatják, és megtehetik ezt az alacsonyabb jogosultsági szinten futó taszk 3 rutinjaival is. Ugyanakkor a taszk 3 a taszk 1, illetve taszk 2 rutinjait csak speciálisan ellenőrzött módon (úgynevezett kapukon keresztül) képes meghívni.

57. Mi a **deskriptor** és mit tartalmaz?

Az azonos privilegizálási szinten futó taszkok tárolóterületét is védeni kell egymástól. Ezért minden egyes taszkhoz speciális védelmi táblázatok kerülnek felépítésre, melyekben található információk (deskriptorok) meghatározzák a taszkhoz tartozó szegmensek és lapokhoz történő hozzáférési jogosultságokat:

- olvasási jog
- írási jog
- végrehajtási jog

58. Az i80x86-os processzorok védett üzemmódja milyen **rendszerobjektumok védelmét** biztosítja? Mire szolgál a **GDT, LDT és az IDT**?

A védelmi rendszer biztosítja a rendszerobjektumok védelmét:

1. program-, verem- és adatszegmensek
2. lapok
3. taszkok
4. I/O portok

GDT = Globális Deszkriptor Tábla (rendszerszintű deszkriptorokat tartalmazza)

LDT = Lokális Deszkriptor Tábla (egy taszkhhoz tartozó deszkriptorokat tartalmazza)

IDT = Megszakítás Deszkriptor Tábla (megszakítás- és kivételkezelő rutinokhoz "vezető" kapukat tartalmazza)

59. Mi a feladata a védelmi rendszerben a **kapuknak**? Milyen kaputípusokat ismer?

A védelmi rendszer 4 fajta kaputípust kezel. Ezek a call, a megszakítás, a trap és a taszk kapu. A call kapuk a különböző privilégiumszintek közötti vezérlésátadásra szolgálnak, a megszakítás, trap kapukkal a hardver-megszakításokat, a processzor által detektált kivételeket (faults, traps, aborts) és a programvezérelt megszakításokat (INT, IRET utasítások) lehet kezelni. Ezek a kapuk egy kódszegmens belépési pontjára mutató címet is tartalmaznak. Taszk kapun történő áthaladás taszkváltást eredményez új TSS-sel. A vezérlésnek kapukon történő „áthaladása” során a privilegizálási szint (PL) mindig ellenőrzésre kerül.

60. Hogy történik a **jogosultság ellenőrzése adathozzáférésnél** az Intel processzoroknál védett üzemmódban?

A jogosultság ellenőrzése a szegmentáláshoz tartozó címfordításnál történik. A kezdeményező legalább olyan magas privilégiummal rendelkezzen mint a cél adatszegmens, ellenőrzésre kerülnek a deszkriptorban levő olvasási, ill. írási jogok.

61. Hogy történik a **jogosultság ellenőrzése vezérlésátadásnál** az Intel processzoroknál védett üzemmódban?

A kezdeményezőnek legalább akkora privilégiuma kell legyen, mint a kapuban levő szint.

62. Mi a feladata a **háttértárolóknak** a **tárolóhierarchiában**?

1. a belső memória kiterjesztését szolgáló virtuális tár kezelése, amely az aktív (futtatás alatt álló) taszkokhoz tartozó programokat és ezek adatait tárolja
2. az aktuálisan nem aktív programfolyamatok programjainak és az ezekhez tartozó adatállományoknak a tárolása - ebbe beleértjük a programok és adatállományok biztonsági másolatainak tárolását is

63. Milyen **részegységekből** épül fel a **háttértároló**?

1. magából az adathordozóból, amely az adatokat és a programokat tárolja - ezek az adatok rögzítésének fizikai megoldását tekintve lehetnek mágneses vagy optikai tárolók
2. az író/olvasó eszközből, mely az adatokat az adathordozóról beolvassa, vagy kiírja
3. a vezérlő egységből, amely a külső tároló működését szervezi, irányítja

64. Fejtse ki a **mágneses jelrögzítés** elvét!

A jelrögzítés alapja az elektromágneses indukció. Minden lemezoldalhoz tartozik egy író-olvasó fej, amely alatt egyenletesen forog a lemez. A fej előre-hátra mozgását a fejmeghajtó-elektronika biztosítja. Írásnál a fej alatt éppen lévő mágneses részecske irányát a fej tekercsében folyó áram határozza meg. Az adattárolás elve szerint a logikai "1" fluxusváltozást okoz, a logikai "0" esetében nincs fluxusváltozás. Olvasásnál a tekercsbe indukált áram minden "1"-nél irányt változtat.

65. Milyen **kódolási eljárásokat** ismer a mágneses adatrögzítéshez?

FM (Frequency Modulation = frekvencia moduláció): Ennél a kódolásnál az alacsony és a magas szinteket eltérő frekvenciájú jelek jelzik. Általában a "0" szinthez alacsonyabb frekvenciájú jel tartozik, mint az "1" szinthez. Ebben az esetben minden bithez tartozik egy szinkronjel, ami kis adatsűrűséget eredményez.

MFM (Modified Frequency Modulation = módosított frekvencia moduláció): Az FM kódolás hiányosságait hivatott kiküszöbölni ez az eljárás, amit úgy értek el, hogy a szinkronjelet csak "0" után hagyták meg. Ezt a kódolást alkalmazták az első merevlemezeknél, de a hajlékonylemezek még ma is ezt használják.

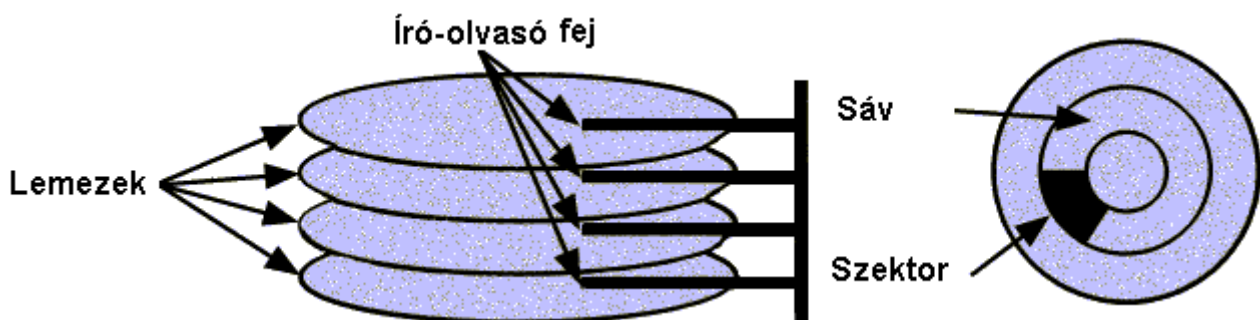
RLL (Run Length Limited) kódolás: Kifejlesztésénél a cél az volt, hogy a fluxusváltozást további 50%-kal csökkentsék. Ezt az adatok átkódolásával sikerült megoldani. A kódolás elve viszonylag egyszerű, két "1" állapot között meghatározott számú "0" állapotnak kell lennie.

66. Melyek egy **merevlemez legfőbb adatai**? Hogyan számítjuk ki a **hozzáférési időt**?

1. Kapacitás
2. Forgási sebesség
3. Lemezátmérő
4. Hozzáférési idő

Hozzáférési idő = Fejmozgatási (pozicionálási) idő átlaga + Forgási idő átlaga + Adatátviteli idő + Vezérlési idő

67. Milyen a fizikai felépítése egy merevlemez egységnek?



68. Mi a **RAID** és milyen megvalósítási formáit ismeri?

A RAID (Redundant Array of Independent Disks) egy olyan mágneslemez-tárolórendszer kiépítést jelent, amely megnöveli a működési biztonságot, és az adatok elérési idejét csökkenti. Ezen célokból egymástól független redundáns lemezegységeket alkalmaz. RAID 0, 1, 2, 3, 4, 5, 6, 0+1, 1+0 létezik.

RAID 0 – összefűzés

RAID 1 – tükrözés

RAID 2 – ECC

69. Miben különböznek a **hajlékonylemez** a merevlemezektől?

1. A merevlemezek kapacitása több nagyságrenddel nagyobb
2. A hajlékonylemez cserélhető adathordozó, míg a merevlemezek zárt, légmentes tokban vannak elhelyezve
3. A hajlékonylemez-egység fordulatszáma nagyságrenddel kisebb a merevlemezénél: 360 fordulat/perc
4. Az adatátviteli teljesítmény is nagyságrenddel kisebb: 500 Kbit/sec;
5. A jelrögzítés a hajlékonylemezeken MFM, a merevlemezeken pedig RLL kódolással történik
6. Az olvasófej fizikailag érintkezik a lemezfelülettel (merevlemezekenél légrés);
7. Az adatátvitel a hajlékonylemez-egységből a DMA-vezérlővel történik, merevlemezekenél a merevlemez-vezérlő irányítja az adatátvitelt (PCI buszos bus mastering)

70. Milyen technológiákat alkalmaznak a **nem hagyományos író-olvasó fejek** kialakításánál?

A hagyományos, elektromágneses indukcióra alapuló fej gyártási technológiáját korszerűsítve kidolgozták a TFI (Thin Film Inductive) író-olvasó fejet. Ez egy parányi tekercset tartalmaz egy vékony filmen. Az 1990-es évek elejétől, a megnövekedett adattárolási sűrűség igénye miatt, az IBM bevezette a mágneses rezisztív, ún MR (Magnetoresistive) fejet, amely két részből áll:

- az író elem egy szokásos TFI-fej
- az olvasó elem egy speciális ferromágneses anyagból (nikkel-vas ötvözetből) készített érzékelő, amelynek az elektromos ellenállása változik az alkalmazott mágneses tér erejének függvényében (ez az ún. anizotrop mágneses rezisztencia effektus)

Az olvasó fej alatt elhaladó mágneses lemez tartalmának megfelelően változik a fejen átvezetett áram erőssége.

71. Mi az **adatok rögzítésének elve a CD lemezek**en?

Az adatok tárolása a visszatükröző réteg bemélyedéseivel (pit), illetve változatlan felületével (land) történik. Valójában a pit és a land is logikai "0"-t jelent, a logikai "1" a kettő közötti átmenetnek felel meg. Az olvasás egy lézersugár visszaverődésének érzékelésével történik.

72. Hogyan helyezkednek el az **adatok** egy **CD** lemezen?

Az adathordozó egy 12 cm átmérőjű és 1,2 mm vastag korong, amelyre az adatok spirál formában kerülnek felírásra. A spirálon az adattárolás szektorokban történik. Egy szektor mérete kb. 2 Kbájt.

73. Mi a sajátossága a több **szessziós formátumnak** a CD-n?

Az újabb meghajtók (CD-írók) lehetőséget adnak az adatok több szesszióban (multi-session) történő felírására, ami azt jelenti, hogy később újabb információt lehet hozzáadni az eddigihez a fennmaradt szabad területen. Ebben az esetben a tartalomjegyzéket aktualizálni kell, és még egyszer felírni.

74. Melyek az **optikai tárolók** legfontosabb **jellemzői**?

1. nagy tárolási sűrűség (10-100-szorosa a merevlemezének)
2. élettartamuk több évtized
3. cserélhető tárolók
4. nagy adatbiztonság
5. elérési idejük relatíve magas (10-100-szorosa a merevlemezének)
6. alacsony adattovábbítási sebesség (audió-CD 1X=150 Kbájt/sec)
7. viszonylag alacsony ár

75. Milyen **típusú optikai tárolókat** ismer?

CD-DA, CD-ROM, CD-R, CD-RW, DVD-ROM, DVD-RAM, DVD-R, DVD-RW, Blu-Ray

76. Miként **szabványosították** az adatok **CD-n** való tárolását?

A CD-ROM-állomány felépítését (fájlformátum) a "zöld könyv" némi módosításával fogadta el 1987-ben a Nemzetközi Szabványügyi Szervezet (ISO = International Standards Organization) mint az ISO 9660-as szabványát.

77. Milyen **feladatokat** lát el egy **CD meghajtója**?

A CD-meghajtónak különböző forrású, szabványos méretű lemezt kell fogadnia, majd a lemezt meg kell forgatnia. A meghajtó lézerfejét fókuszálni kell a lemezre, és a fejnek nagy pontossággal (1 mikrométeren belül) kell követnie a lemezre írt spirális sávot. A meghajtó-elektronika feladata az olvasott adatok felismerése, a sorosan érkező csatornabitek összeállítása és visszakódolása bájtokra, az előforduló olvasási hibák észlelése és javítása, az olvasott bájtok átadása az interfész felületen a CD-vezérlőnek, valamint a hang CD-k jelének felerősítése a fejhallgató kimenetre.

78. Melyek a **CD-meghajtó optikai rendszerének a feladatai** és azokat hogy valósítják meg?

Az optikai rendszer két fő feladatot lát el:

1. a CD-n tárolt (lyukak formájában rögzített) információ digitális jelekké alakítása
2. a spirálisan kanyarodó felírás pontos követése az olvasófej által

Mind az adatok olvasását, mind a sávon tartást az optikai rendszerből sugárzott és a lemezeről a lyukakról a fejbe visszaverődő lézersugárral oldották meg.

79. Hogyan értelmezzük a **CD-ROM-meghajtók** gyárilag megadott **sebességét**?

Alapsebességnek az audió CD sebességét tekintjük, a CD-ROM-ok sebességét az ehhez viszonyított szorzószámmal fejezzük ki. Az audió CD sebessége 150 Kbájt/s, ez az 1X sebesség.

80. Mi a **DVD** és miben különbözik a CD-ROM-tól?

A név kezdetben Digital Video Disc-et jelentett, később a Digital Versatile Disc (sokoldalú digitális lemez) használata terjedt el.

1. a szomszédos sávok közötti kisebb távolság
2. kisebb méretű lyukak alkalmazása
3. két, egymásra helyezett adathordozó réteg lehetősége (a felső félig átlátszó)
4. kétoldalas lemez kialakításának lehetősége (két 0,6 mm vastagságú lemez összeragasztásával)
5. hatékonyabb logikai formátum (rövidebb hibajavító mező)

81. Mi a **flash memória** és milyen eszközökben használják?

A flash memória olyan törölhető és újraírható félvezető tároló, amely az adatokat tápellátás hiányában is megőrzi. Az elektromos úton törölhető EEPROM memóriáknak egy továbbfejlesztett változata, az a sajátossága, hogy a törlést és újraprogramozást blokkonként teszi lehetővé, ami jelentős sebességnövekedést okoz. A pendrive-ok és a memóriakártyák és SSD meghajtók a flash memóriát használják az adatok tárolására.

Kommunikáció a számítógép részegységei között

1. Sorolja fel a **számítógép fontosabb részegységeit!**

1. a processzor
2. az L2 cache
3. a főtár
4. a megszakításrendszer
5. a közvetlen memóriáhozáférési (DMA) rendszer
6. I/O eszközvezérlők és interfészek
7. a perifériák
8. a háttértárak
9. a busz- vagy sínrendszer

2. Egy egérekattintással elindítunk egy programot. Ismertesse a feladat során együttműködő hardveregységeket az animáció alapján!

1. A felhasználó elindít egy "NYOMTATÓ" nevű programot, azzal, hogy a képernyőn egérrel rákattint a program nevére. Ennek hatására az operációs rendszer a merevlemezről betölti a programot a RAM-ba.
2. A program megkezdí futását: elsőként egy szöveget vár, amelyet a billentyűzetről kell begépelni. A "NYOMTATÓ" nevű program a beolvasott szöveget átvizsi a RAM-ba.
3. A program a RAM-ban tárolt "Jó reggelt!" szöveget kiírja a monitor képernyőjére.
4. Miután a felhasználó elfogadja hibátlanak a képernyőnmegjelenő szöveget, egérekattintással parancsot ad a nyomtatásra.
5. A "NYOMTATÓ" program a RAM-ból kinyomtatja a begépelte "Jó reggelt!" szöveget, amely megjelenik a nyomtatóban.

3. Milyen típusú események kezeléséhez kell **átmenetileg felfüggeszteni** a **program futását**?

1. meghatározott programhibák esetén (így például a program egy aritmetikai műveletben 0-val akar osztani)
2. meghatározott műveletek befejezésekor, amelyek bekövetkezésére számítani lehet, de ezek időpontja pontosan nem tervezhető (például amikor egy periféria jelzi, hogy egy input vagy output műveletet befejezett)

3. szándékosan, azaz programvezérelt módon (amikor például a programhibák felderítése céljából a programfutást lépésenként megszakítjuk annak érdekében, hogy megtekintsük a tárolótartalmakat)
4. teljesen véletlenszerűen és váratlanul (ilyen például egy súlyos hardverhiba vagy áramkimaradás)

4. Mi a különbség a **megszakítás** és a **kivétel** között? Mit jelent a **maszkolható** és **nem maszkolható megszakításkérelem**?

1. a programfutáshoz képest külső eredetű megszakítások (interrupt) (például egy I/O eszköz adatátviteli igényének jelzése)
 2. az utasítások szabályszerű végrehajtását megakadályozó kivételek (exception), amelyet a processzor egy utasítás végrehajtása során észlel
- olyan események, amelyek megszakítási igénye átmenetileg letiltható - ezeket maszkolható megszakítási kérelmeknek nevezik
 - olyan események, amelyek megszakítási igénye nem tiltható le és minden esetben ki kell szolgálni - ezeket nem maszkolható megszakításoknak, azaz NMI = Non Maskable Interrupt-oknak nevezik. (Ilyenek például a súlyos hardverhibák.)

5. Mi a lényege a **vektoros megszakítási** rendszernek?

A megszakításkérelmeket ún. vektoros módon dolgozzák fel, ami azt jelenti, hogy a megszakításkérelem a megszakításkiszolgáló rutin kezdőcímét egy címekből álló vektor elemeként azonosítja a processzor számára.

6. Sorolja fel a **megszakítás-kiszolgálás hardver-** és **szoftverlépéseit!**

A hardver által:

1. a megszakításvezérlő beállítja a megszakításkérő vezérlő vonal jelszintjét, ezzel jelzi a processzornak a megszakításkérelmet (INT jel)
2. a processzor a folyamatban lévő utasítást befejezi, majd visszaigazolja a megszakításkérelem elfogadását (IACK jel)
3. ezt követően a megszakításvezérlő a sínre küldi a megszakítási vektor elemének sorszáma
4. a processzor tárolja a megszakítási vektor elemének sorszáma
5. a processzor elmenti a verembe az utasítászámláló és az állapotregiszter tartalmát
6. a processzor a megszakítási vektor elemszáma alapján a megszakításkiszolgáló rutin kezdőcímét betölti az utasítászámláló regiszterbe és ezzel megkezdődik a megszakítás kiszolgáló rutin végrehajtása

Az operációs rendszer által:

1. a megszakított program adatainak (regisztereinek) elmentése verembe (ha szükséges)
2. a megszakítás okának behatárolása
3. a kiszolgáláshoz szükséges adatok összegyűjtése
4. a megszakítást okozó esemény kezelése
5. a megszakított program adatainak visszatöltése
6. a megszakítás kiszolgáló rutin befejezésének jelzése

A hardver által:

1. az elmentett állapot és utasítászámláló regiszter tartalmának visszatöltése és a megszakított program folytatása

7. Mi a feladata a megszakításvezérlőnek?

1. fogadja megszakításkérő vezérlővonalakon (IRQ = Interrupt ReQuest) a megszakításkérelmeket
2. vizsgálja, hogy az igényelt megszakítás nincs-e maszkolással letiltva
3. vizsgálja és értékeli a megszakítás prioritását
4. az INT (Interrupt) vezetéken közli a megszakítás kérést a processzorral
5. ha az IACK (Interrupt Acknowledgment) vezetéken a processzor visszaigazolja, hogy kész a kérés fogadására, akkor a megszakításvezérlő átadja a processzornak a megszakításhoz tartozó megszakításvektor címet

8. Mi a DMA lényege?

A közvetlen memóriahozzáférés lényege, hogy a processzor egy I/O művelet végrehajtásához szükséges információkat átadja egy, a processzortól független DMA (Direct Memory Access) vezérlőnek, amely ezt követően az adatátvitelt a memória és az I/O eszköz között önállóan irányítja. Ezáltal a processzor felszabadul más feladatok végrehajtására.

9. Ismertesse a DMA regisztereit és ezek feladatait!

1. DMA-módregiszter, amely az adatátvitel irányára (memóriába írás, vagy memóriából történő olvasás) vonatkozó információkat tartalmaz
2. DMA-maszkregiszter, amely az egyes DMA átvitelt kérő vezérlővonalak letiltását (maszkolását) tartalmazza
3. DMA-állapotregiszter, amely a vezérlő állapotával kapcsolatos információk tárolására szolgál

10. Milyen regisztereket tartalmaznak az I/O eszközvezérlők és ezeknek mi a feladata?

1. parancs (command) regiszter, amely az eszközvezérlő által végrehajtandó műveletekhez szükséges információkat tárolja
2. állapot (status) regiszter, amelyben az eszközvezérlő az I/O eszköz aktuális állapotára vonatkozó információkat tárolja(pl. egy merevlemezre egy blokk kiírása megkezdődött, vagy a nyomtatóból kifogyott a papír)
3. az adatkiírás illetve beolvasás pufferregiszterei, amelyek a folyamatban lévő I/O műveletek adatait tárolják

11. Hogyan irányítja a processzor az I/O eszközvezérlőket?

1. közvetlen I/O utasításokkal (miután az állapotregiszter lekérdezésével megállapította, hogy az eszköz az utasítás végrehajtására képes állapotban van), a parancsregiszter beállításával és a pufferregiszterek írásával vagy olvasásával
2. közvetett módon (memória-átviteli utasításokkal), amikor a címzés úgy történik, mintha az I/O eszköz regiszterkészlete a főtár része lenne

12. Mi a feladata a buszrendszernek és a buszvezérlőnek?

A számítógép részegységei közötti kommunikációs kapcsolatokat (adatok, címek, valamint a gép vezérléséhez szükséges információk átvitelét) a sín- vagy buszrendszer biztosítja. A sínprotokollban meghatározott szabályrendszer fizikailag a buszvezérlő hardveregységben testesül meg.

13. Az átvitt adatok jellege szerint milyen **logikai részekből** tevődik össze a **sínrendszer**?

1. címsín, amely a címek átvitelét biztosítja. A processzor címkezelésének (pl. 32 vagy 64 bites processzorok) megfelelően általában 32 vagy 64 címvezetékkel artalmaz
2. adatsín, amely az adatok átvitelét biztosítja, szélessége általában 32 vagy 64 bit
3. vezérlősín, amely a számítógép részegységei között a vezérlőinformációk adatátvitelét biztosítja

14. Sorolja fel a sínrendszeren átvitt legfontosabb **vezérlőjeleket**!

1. adatátvitelt, azaz az I/O eszközöket vezérlő jelek
2. a megszakítási rendszerhez tartozó vezérlőjelek
3. a DMA vezérlőjelei
4. a sínvezérlőjelek (pl. a sínhasználat kérése és ennek visszaigazolása)
5. szinkronizációs jelek

15. Jellemezze a **belső és külső sínrendszert**, valamint ez utóbbi fajtáit!

1. belső sínrendszer - amely a processzoron belül, a processzor különböző részeit kapcsolja össze, sebessége (órajele) megegyezik a processzoréval
2. külső sínrendszer - amely a processzort köti össze a központi egység különböző részegységeivel

A külső sínrendszer a sebessége és az összekapcsolt eszközök alapján több, hierarchikusan felépülő fajtájú lehet:

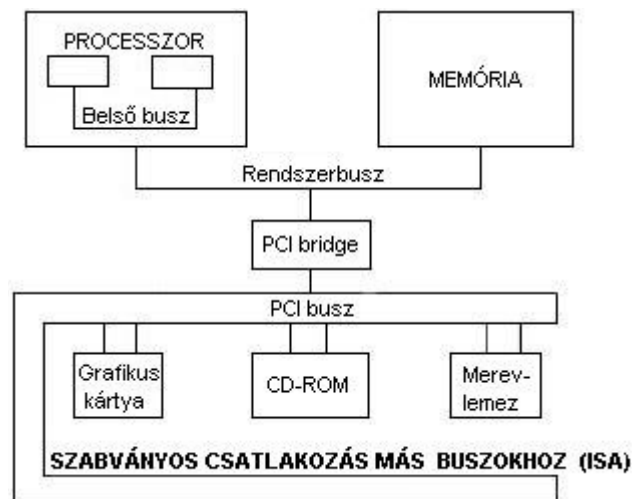
1. rendszersín (system bus), amely a processzorhoz közvetlenül kapcsolódó rendszerelemeket (memória, grafikus kártya stb.) köti össze. Ezen keresztül az adatátvitel a processzor órajelével szinkronban történik és a busz adatátviteli bit szélessége is összhangban van a processzor működésével (32-bites processzoroknál 32 vagy 64 bit)
2. I/O-sín, amelyet egy sínvezérlő egység hajt meg, és alapvetően az I/O eszközök csatlakoztatását szolgálja
3. az egyéb I/O eszközök saját sínrendszere (például SCSI lemezcsatoló önálló busza)
4. számítógéprendszerek közötti buszok (intersystem bus)

16. Mi a feladata a **rendszer- és az I/O-busznak** a PC-ben?

1. a rendszerbusz (system bus), amely a processzort köti össze a rendszermemóriával (L2 cache és DRAM modulok)
2. az I/O busz, amely a periféria- és háttértár-vezérlésekkel biztosítja a kapcsolatot

17. Mi a **PCI sín** létrejöttének alapgondolata? Váolja fel a PCI sín **blokkvázlatát**!

PCI (Peripheral Component Interconnector) ipari szabvány. Ennél a rendszersín és a PCI busz közé egy processor-PCI Bridge-t iktattak be, amely lehetővé tette, hogy a buszrendszer a konkrét processzortól és annak sebességétől függetlenül is működőképes legyen. Ez azt jelenti, hogy a PCI busz nem kötődik szorosan egy processzorhoz, hanem többfajta processzorral is képes együttműködni.



18. Jellemezze röviden a **PCI Express**-t!

A PCI Express sínrendszer soros átvitelen alapuló, pont-pont busztopológiával működik, vonalkapcsolatonként 200 Mbájt/s-os sebességgel (több vonal létrehozásával ez a szám növelhető).

19. Mi a **buszciklus**, a **busztranzakció**, és milyen lépései vannak?

Buszciklus: egy adategység átviteléhez szükséges idő (a ciklusok végén jelennek meg a szükséges jelszintek).

Busz tranzakció: a buszigények sorozata, amely az adatátvitel igénylésétől, annak befejezéséig tart (több műveletből, buszciklusból állhat).

Egy busztranzakció lépései a következők lehetnek:

1. sínhasználat igénylése (REQUEST)
2. sínhasználat jogának odaítélése (ARBITRATION)
3. címátvitel (ADDRESSING)
4. adatátvitel (DATA TRANSFER)
5. hibafelismerés (ERROR DETECTION)
6. felszabadítása (RELEASE)

20. Milyen fontosabb **adatátviteli típusokat** ismer a buszon?

1. egyes átvitel (minden címzés után egy adat kerül átvitelbe)
2. burst átvitel (egy címzés esetén több - folyamatos címeken lévő - adat átvitele valósul meg)

21. Definiálja a **master** és **slave** eszközöket és legfontosabb feladataikat!

A sín egyidőben csak egy eszközpár használhatja. Ezek közül az aktív kezdeményező eszköz a master, a kapcsolatban résztvevő passzív eszköz pedig a slave eszköz. A master elindít és befejez egy busztranzakciót, címet küld. A slave válaszol az igényekre és címekre, a sínre teszi illetve fogadja az adatokat.

22. Mi a **sín arbitráció**, milyen típusú buszok vannak?

Egyidejűleg több aktív (master) eszköz is igényelheti a sín használatát. Ekkor valamilyen eljárással el kell dönteni, hogy melyik eszköz lesz jogosult a sínhasználatára. Ezt az eljárást nevezzük sín arbitrációnak.

Egyes busztípusok jellemzői:

processzor – tároló közti busz

- architektúra specifikus
- rövid, nagy sebességű

I/O busz

- ipari szabvány,
- hosszabb és lassúbb a processzor – tároló busznál, és több különböző egységet szolgál ki. Buszadapterrel kapcsolódik a processzor – tároló buszhoz.

Processzor – tároló – I/O eszközök egy buszon

- olcsó
- lassú (összességében)
- ma már ritka

23. Hogyan lehet **címezni** a **buszrendszerre** rákapcsolt eszközöket?

Alapszabály, hogy mindig a master címzi a slave-t. Ez a címzés lehet:

1. helyfüggetlen (minden eszköznek egyértelmű logikai címe van)
2. memory mapped addressing (az eszköz címzése úgy történik, mintha a főtár része lenne)
3. broadcast (az összes slave megcímzése egyidejűleg megtörténik pl. inicializáláskor)

24. Milyen lépésekből áll az egyszerűsített **handshaking**?

Az „ADÓ” egy vezérlőjellel informálja a „VEVŐ”-t, hogy az érvényes (elküldendő) adatokat az adatsínre helyezte (DAV = Data Valid = érvényesek az adatok vezérlőjel = 1).

A DAV jel 1-es állapotát érzékelve a „VEVŐ” olvassa az adatbusz adatait, és ennek hibátlan megtörténtét egy DAC = Data Accepted = az adatok fogadása megtörtént vezérlőjel = 1-es szintre állításával jelzi az „ADÓ”-nak.

25. Mit jelent a **fully interlocked handshaking**?

1. Az ADÓ az első adatbájtot a sínre helyezi.
2. Az ADÓ a DAV jelet 1-re állítja.
3. A VEVŐ visszaigazolja az adatok átvételét (DAC = 1).
4. Az ADÓ visszaállítja a DAV jelet 0-s szintre.
5. A DAC jel 0 szintre állításával a VEVŐ jelzi, hogy felkészült az újabb adat fogadására.

26. Miért kell „**timer**”-t alkalmazni a kézfogósos algoritmusban?

Ezzel figyelhető, hogy a DAV jel visszaigazolása egy meghatározott időn belül megtörténik-e, és ha nem, akkor ez egy megszakítást eredményez, és egy hibakezelő rutin futását elindítja.

27. Jellemezze a **szinkron sínvezérlést**!

A szinkron sínvezérlés esetén az eseményeknek rögzített időpontjaik vannak, a sínen kommunikáló eszközök azonos órajellel ütemezettek. Az adás-vétel mindig azonos sebességgel történik, nem kell kapcsolatfelvétel és visszaigazolás. A szinkron sínvezérlés előnye gyorsasága, hátránya viszont, hogy közös órajelet kell biztosítani az összes sínre kapcsolt eszköz számára.

28. Jellemezze az **aszinkron sínvezérlést**!

Aszinkron sínvezérlésnél az események tetszőleges időpontban bekövetkezhetnek. Ezért a sínre csatlakozó eszközök zavartalan együttműködéséhez egy kapcsolatfelvétel és vétel visszaigazolási eljárás (handshake) szükséges.

Az aszinkron sínvezérlés előnye, hogy nem kell azonos órajelet biztosítani az összes sínre kapcsolt eszköz számára, és nagyon eltérő sebességű eszközök kiszolgálását is lehetővé teszi. Hátránya viszont, hogy a protokollba be kell építeni a relatíve bonyolult aszinkron „handshake” eljárást.

29. Mit jelent a **blokkos** és a **megosztott adatátvitel** a sínen?

Blokkátvitel

1. Egy tranzakcióban több memóriaszó átvitelére kerül sor
2. A címet csak egy alkalommal kell kiküldeni
3. A busz csak akkor kerül felszabadításra, ha az utolsó szó is átvitelre került

Megosztott átvitel (a példában egy I/O eszköz adatot olvas a tárolóból)

1. Az I/O eszköz küldi az adatátviteli igényt és a címet, azt a tároló visszaigazolja, azt követően a busz felszabadításra kerül.
2. A tároló jelzi, hogy az adatok átvitelre készen állnak. Az I/O eszköz visszaigazolja az adatok átvételét, majd a busz felszabadításra kerül.

30. Mit jelent a **statikus** és **dinamikus buszhasználati** jog megosztás?

A buszhasználat jogának megosztása a mesterek között a legegyszerűbb módon időosztással (time-sharing) történhet, amikor minden master meghatározott időszelvre megkapja a buszhasználat jogát (statikus módszer). Csak akkor hatékony, ha a mesterek adatátviteli igénye kb. azonos.

Dinamikus buszhasználat megosztás esetén a mesterek csak akkor kapják meg a sít, ha azt igénylik. Ez esetben azt a problémát kell kezelni, hogy a buszhasználati igények azonos időpontban is jelentkezhetnek.

31. A lefoglalt **busz felszabadításának** melyek a legfontosabb módszerei?

1. release on request: a master annyi időre lefoglalja a buszt (akkor is, ha adatot nem forgalmaz), amíg a buszt másik master nem igényli. Ezt alkalmazzák leggyakrabban mikroszámítógépek buszrendszereiben
2. release when done: a master egy tranzakcióra kapja meg a buszhasználat jogát, annak befejezése után felszabadítja a buszt
3. preemption: ha egy magasabb prioritású master jelentkezik, ez a tranzakciót megszakítja. Ezt blokküzem módban van értelme alkalmazni

32. Mi a **dinamikus buszhasználat-megosztás** feltétele?

Dinamikus buszhasználat-szétosztás csak akkor lehetséges, ha:

1. minden master jelzi a buszfoglalási igényt
2. egy master csak akkor használhatja a buszt, ha igényének elfogadását visszaigazolja
3. a masternek jelzést kell küldenie, ha a tranzakció befejeződött

33. Mi a busz arbiter? Határozza meg a központosított és szétosztott **buszarbitrációt**!

Egy olyan hardver egység, amely a sínfoglalási kérelmeket fogadja, elbírálja és visszaigazolja.

1. központosított, ekkor a számítógéprendszerben csak egy arbiter van. Ez lehet egy önálló hardveregység, vagy egy másik hardvereszköznek (legtöbbször a CPU-nak) valamilyen részegysége
2. szétosztott, akkor a számítógéprendszerben több arbiter hardver egység található (ez multiprocesszoros rendszerek esetén gyakoribb)

34. Jellemezze a **soros sínfoglalást**!

Soros kiszolgálás esetén az eszközök sorba vannak kötve, és sorrendjük határozza meg prioritásukat. A buszprotokoll lefutása soros sínfoglalás esetén 1. Request (kérés); 2. Grant (engedélyezés); 3. Bustransaction (tranzakció-lebonyolítás); 4. Release (felszabadítás). A hely szerinti prioritás azzal érvényesül, hogy a "Grant" jelet a felfűzött eszközök továbbengedik az első igényig, amely azt "elfogván" megkapja a buszhasználat jogát. A soros sínfoglalás előnye egyszerűségében van. Hátránya, hogy a kis prioritású eszközök sokáig várhatnak.

35. Jellemezze a **párhuzamos sínfoglalást**!

Párhuzamos kiszolgálás esetén minden sínhasználatért folyamodó eszköz önálló kérő és engedélyező vezérlővonallal rendelkezik, és a busz arbiter prioritás szerint engedélyezi a sín igénybevételét. A prioritás meghatározása különböző eljárások szerint történhet, például:

1. az egyszerű körbejáró eljárásnál minden sínhasználatot követően a korábban legalacsonyabb prioritású eszköz kapja meg a legmagasabb prioritást, az összes többi egyvel alacsonyabb prioritási szintre kerül
2. az LRU (Last Recently Used) eljárásnál az az eszköz kapja meg a sínhasználat jogát, amely a sít legrégebben vette igénybe. Ezekkel az eljárásokkal az eszköz esélyei a buszhasználat elnyerésére időben kiegyenlítődnek

36. Ismertesse a rendszerbusz gyorsított (**burst**) átviteli módját PC-kben!

Egyetlen cím elküldésével egy cache-adatsort, azaz $4 \times 8 = 32$ bájt méretű blokkot másol be a cachebe, vagy ír vissza a memóriába. Ha a sín szélessége 64 bit, akkor a rendszersínen egy órajel alatt 64 bit = 8 bájtnyi adatot vagy a memóriára vonatkozó címinformációt lehet átvinni.

37. Milyen esetekben működik a rendszerbusz **egyes átviteli módban**, és ez esetben hogyan történik meg az adatátvitel?

Az I/O címtartományba és a rendszermemória cachetárral nem gyorsítható részébe (például a monitorvezérlő kártya frame buffere esetén) eső címek esetében a Pentium rendszerbusza egyes átviteli módban működik. Ekkor a címátvitelt követően a sín az adatokat 64 bit = 8 bájtos egységekben továbbítja. A cachevezérlőnek egy vezérlőjel magas szintje jelzi, hogy nem kell az átvitt adatblokkal a cache egy sorát feltölteni, ennek hatására az adatok egy regiszterbe kerülnek.

38. Mi a szerepe a **bájt kiválasztó** vezérlőjeleknek?

Ezek a jelek határozzák meg, hogy az adatbusz melyik 8 bites csoportján zajlik az átvitel. Ezekkel a processzor a memória egyedi bájtjaihoz, szavaihoz és dupla szavaihoz is hozzáférhet.

39. Mi a **pont-pont** összeköttetésen alapuló **kommunikációs** kapcsolatok lényege?

Az eszközök - végpontok formájában - egy közös kapcsolóhoz csatlakoznak, amely prioritásuk szerint továbbítja az adatokat a CPU és az eszközök között. Mindegyik eszköz saját soros vonallal rendelkezik, ami lehetővé teszi az egyidejű működést. A vonalak száma növelhető, így biztosítva nagyobb átviteli rátát, ha szükséges. Az adatok továbbítása csomagok formájában történik, megfelelő irányításuk a kapcsoló feladata. Az architektúra a számítógép-hálózatokból ismert OSI-rétegekhez hasonlóan épül fel. A legfelső szoftverszint fogadja a hívásokat, a CPU címtérületének megfelelően. A következő szint felel az adatcsomagok szállításáért, majd az adatkapcsolati réteg hivatott az adatok sértetlenségét biztosítani hibajavító kódolással. A legalsó, fizikai szint látja el az adatbitek differenciális feszültségszintekkel történő átvitelét.

40. Adja meg a **tranzakció-orientált** I/O, a **fájl** I/O és az **eseményorientált** I/O fogalmát és teljesítményjellemzőit!

Tranzakcióorientált I/O jellemzője a sok, kis változtatás nagy adathalmazokban (pl.: bankautomaták, repülőgép-helyfoglalás). A számítógéprendszer teljesítménye szempontjából a legfontosabb az időegység alatti hozzáférések száma.

Fájl I/O jellemzője a soros hozzáférés nagy adathalmazokhoz. Emiatt a legfontosabb az időegység alatt átvitt adatmennyiség.

Eseményorientált I/O esetében a számítógépes rendszernek külső eseményekre kell reagálnia (pl.: egér helyzetének megváltoztatása, folyamatvezérlés). Emiatt e rendszerekben a reakcióidő, valamint az időegység alatti feldolgozható eseményszám a legfontosabb teljesítmény jellemző.

41. Mitől függ a számítógép I/O **teljesítménye**?

1. a válaszidő (azaz egy I/O igényt a számítógéprendszer mennyi idő alatt képes kiszolgálni)
2. az adatátbocsátó képesség (pl. Mbájt/sec-ben mérve)

42. Milyen I/O **műveletekkel** összefüggő feladatai vannak az **operációs rendszernek**?

1. A védelmi funkció: A taszkok által közösen használt I/O eszközöket a felhasználói programok csak felszabadítást követően vehetik igénybe, ezt az operációs rendszernek ellenőriznie kell.
2. Eszközspecifikus szolgáltatások
3. Az I/O műveletek hibakezelésének egységes megoldása.

43. Az I/O **eszközvezérlők** milyen esetekben és formában adnak át **információkat** az operációs rendszernek?

Az operációs rendszert informálni kell:

1. ha az I/O eszköz befejezett egy műveletet
2. ha az I/O művelet hibátlanul végrehajtásra került

Ez kétféle képpen történhet:

1. Polling alkalmazásával (azaz az eszközök állapotának „körbekérdezésével”):
Ez esetben az I/O egység az állapot (státus) regiszterben kódolja az átadandó információt, az operációs rendszer pedig periodikusan és ismétlődően lekérdezi a státuszregiszter bitjeit, hogy az eszköz I/O műveletre való alkalmasságát vagy az adatátvitel hibátlan lefutását megállapítsa
2. I/O megszakítás alkalmazásával:
Ez esetben az I/O eszköz a megszakítási rendszer segítségével informálja az operációs rendszert az I/O eseményekről.

44. Mutassa be az I/O eszközök **címzésének** lehetőségeit!

1. közvetlen I/O címekkel, ekkor az I/O utasítások a processzor utasításkészletében szerepelnek (ez például az INTEL processzorokra jellemző)
2. közvetett módon (memory mapped addressing), amely esetben a címzés úgy történik mintha az I/O eszköz regiszterkészlete a főtár része lenne (ez a RISC processzorokra jellemző).

45. Definiálja az I/O **adatátvitel** típusait!

1. Programozott I/O átvitel. Ekkor az adatátvitel az I/O eszközök és a főtár között csak a processzor közbeiktatásával (vezérlésével) történhet meg.
2. Megszakításos I/O átvitel, amikor a processzor jelzi az I/O eszköz számára az adatátviteli igényt, amely ha felkészült az adatátvitelre, akkor ezt egy megszakításkérelemmel jelzi a processzornak. Az adatátvitel befejezését az I/O eszköz szintén megszakításkérelemmel jelzi. (INTEL processzoroknál a megszakítási vektorban található meg az I/O eszközt kiszolgáló rutin címe.)
3. Közvetlen memória-hozzáférés, amikor az I/O eszköz és a főtár közötti adatátvitelt a processzortól „függetlenül” a DMA-vezérlő irányítja. A processzor feladata az I/O előkészítése (az I/O eszköz állapotának vizsgálata) és az I/O művelet hibátlan végrehajtásának ellenőrzése. (Ez esetben a processzor és a DMA-vezérlő a kapcsolatot a megszakítási rendszer segítségével tartja fenn.)
4. I/O processzor alkalmazása (főleg mainframe-ekre jellemző), amelynek a processzor átadja az I/O művelet végrehajtásához szükséges összes adatot, és ezt követően az I/O processzor teljesen önállóan vezérli az I/O művelet végrehajtását.

46. Jellemezze a **programozott I/O** átvitelt!

A processzor teljeskörűen ellenőrzi és vezérli az I/O műveletet. A periféria állapotregiszter ciklikus lekérdezése folyamatosan terheli a processzort. Ezért ennek a módszernek csak akkor van értelme, ha az I/O eszköz nagyon gyors.

47. Hogyan zajlik az **adatátvitel** a mágneslemezekenél **PIO Mode**-ban?

A processzor vezérli az I/O műveleteket a lemezvezérlő vezérlőregisztereinek írásával, az adatregiszterek olvasásával/írásával, valamint hiba- és állapotinformációk kiolvasásával. Az adatokat a lemezvezérlő adatregisztere pufferelemi.

48. Jellemezze a **megszakításos I/O** átvitelt!

A felhasználói program a tranzakció idejére megállítássalra kerül. Speciális hardverigény: megszakítási kérelem generálása (I/O eszköz), megszakítás felismerése (processzor, megszakítás-vezérlő), megszakítás-kezelés (processzor).

49. Hogyan alkalmazzák a **megszakításos I/O** átvitelt a PC-k **billentyűzete** esetében?

A billentyűzet egy gombjának lenyomását egy vezetékekből álló mátrix (Scan Matrix) érzékeli, amelyet egy, a billentyűzetbe beépített mikroprocesszor önállóan felügyel. Ez pufferelemi a lenyomott billentyűzetgombnak megfelelő "scan"kódot, majd azt a csipkészlet (alaplapi áramkörkészletbeli) billentyűzet adatregiszterébe továbbítja aszinkron bitsorozat formájában. Minden egyes billentyű lenyomása megszakítást generál, amely alapján elindul az operációs rendszernek megfelelően, az adatregiszter feltöltése egy megfelelő billentyűzet-megszakításkezelő rutinja. Ez kiolvassa a vezérlőben lévő adatregiszterből a "scan" kódot.

50. Jellemezze a **DMA I/O** átvitelt!

A DMA által vezérelt I/O műveletek célja a processzor tehermentesítése. Ekkor a DMA a buszmaster, és a processzortól „függetlenül” vezérli az adatátvitelt.

51. Milyen lépései vannak a **DMA adatátvitelnek**?

1. a processzor ellenőrzi a perifériát, hogy tudja-e fogadni az átvitelt, ezt követően a DMA vezérlő részére átadja az átvitel paramétereit
2. a DMA buszfoglalási kérelmet jelez (DMA REQUEST), a processzor ezt visszaigazolja (DMA ACKNOWLEDGE)
3. DMA masterként lefoglalja a buszt, végrehajtja az adatátvitelt
4. a DMA jelzést küld megszakítással a processzornak az átvitel befejezéséről
5. processzor ellenőrzi a végrehajtás hibátlan megtörténtét, és a buszengedélyezést megszünteti

52. Hogyan történhet a **buszhasználat** megosztása a processzor és a DMA vezérlő között?

Blokkátvitel (blocktransfer) esetén a DMA vezérlő az adatátvitel teljes idejére lefoglalja a buszt.

Cikluslopás (cycle stealing) esetén busz felszabadítása minden szó (bájtt) átvitel után megtörténik. Ez átlapolat buszhasználatot jelent a processzor és a DMA vezérlő között. Ütközés esetén a DMA-nak prioritása van.

Transzport mode esetén a DMA vezérlő használja a buszt (például a processzor utasítás dekódolása és végrehajtása alatt) addig, amíg a processzor nem végez memóriáhozáférést. Ez a mainframe-kre jellemző.

53. Írja le röviden egy **lemezegységgel** történő adatátvitel menetét **Ultra DMA**-módban!

Az Ultra DMA-módban a processzor csak elindítja a burst adatátvitelt. Ez négy fázisban történik: inicializálás, átvitel, szünet, befejezés.

54. Mi az **I/O processzor alkalmazásának** lényege?

Az I/O processzorok (Multiplex, Szelektor és Blokkmultiplex csatornák) az I/O adatátvitel teljes vezérlését önállóan megoldják, tulajdonképpen társprocesszorként működnek. „Felprogramozásukat” a processzor végzi azáltal, hogy átadja az I/O eszközök „csatornaprogramját” az I/O processzor számára (CCW = Channel Command Word, CAW = Channel Address Word).

3. ZH

SZEMÉLYI SZÁMÍTÓGÉPEK

A PC fogalma és fontosabb jellemzői

1. Mit jelent és honnan származik a **PC elnevezés**?

A személyi számítógép, angolul Personal Computer, azaz rövidítve a PC elnevezés azt követően vált általánossá, hogy az IBM az 1981-ben bemutatott, személyes használatra szánt, egy mikroprocesszort tartalmazó számítógépét PC néven vitte piacra.

2. Határozza meg a **PC fogalmát**!

Napjainkban PC alatt olyan relatíve kisteljesítményű, kis méretű, és viszonylag olcsó számítógépet értünk, amelyet nagy sorozatban gyártanak, működésük valamilyen Intel vagy azzal kompatibilis processzorra épül, operációs rendszerként a Windows, Unix stb. különböző verzióinak használata jellemez.

3. Milyen tendenciák jellemzik a következő években a **PC-k piacát**?

Legdinamikusabban a táblagépek száma emelkedik, ezt követik a laptopok, notebook-ok, míg az asztali PC-k részaránya a 2010-es évektől kezdve csökken.

4. Milyen okai vannak a PC-k gyors **elterjedésének**?

Az integrált áramkör-technológia fejlődése egyre inkább lehetővé tette a PC-k gazdaságos gyártását. A PC olcsósága gyorsan bővítette felhasználásának szakterületeit, ez egyúttal újabb keresletet teremtett, és az ebből fakadó nagyobb gyártási sorozatok ismét alacsonyabb árszintet eredményeztek. A PC ipari szabvánnyá válása és az eredeti PC-vel való program-kompatibilitásnak a teljes fejlődési folyamat során való megőrzése jó feltételeket teremtett a szoftverek fejlesztésére.

5. Mit jelent és mi a szerepe az **X86-os utasításkészletének**?

A PC-k lefelé való program-kompatibilitásának biztosításához a legfontosabb eszköz az ún. x86-os utasításkészlet volt, melyhez tartozó utasításokat minden PC processzora ismer és képes végrehajtani. Azoknak az utasításoknak az összességét, amelyek nem, vagy lényegében csak a regiszter operandusok hosszában változtak a 8086-os processzor megjelenése óta, x86-os utasításkészletnek nevezzük.

6. Milyen **funkcionális részegységei** vannak a PC-nek?

1. az adatfeldolgozást végző, valamint a részegységek összehangolt működését vezérlő egység, amely maga a szűkebb értelemben vett számítógép
2. az adatok átmeneti vagy huzamosabb ideig történő tárolására szolgáló eszközök
3. az adatbevitelre szolgáló eszközök vagy input egységek
4. az adatmegjelenítésre szolgáló eszközök vagy output egységek
5. a kommunikációs kapcsolatokra szolgáló eszközök vagy input/output egységek

7. Milyen részekből áll egy PC-s munkahely **alapkonzfigurációja**?



8. Mit nevezünk **perifériának**?

A számítógépek adatbevitelre/adatkivitelre és a kommunikációs kapcsolatokra szolgáló részegységeit összefoglaló néven perifériáknak nevezzük.

9. Sorolja fel a PC **fontosabb perifériáit**, és feladatukat!

1. A PC billentyűzetének segítségével a felhasználó begépeléssel parancsokat, utasításokat és szöveges adatokat vihet be a számítógépes rendszerbe.
2. Az egér mozgatásával a felhasználó a kurzor helyzetét vezérelheti a képernyőn, illetve az itt megjelenő grafikus formátumú menükre kattintva utasításokat közölhet az operációs rendszerrel és a futó programokkal.
3. A monitor szöveges adatok, képek és mozgóképek megjelenítésére szolgáló eszköz.
4. A nyomtató segítségével lehet a számítógépes feldolgozás eredményeit papíron rögzíteni.

10. Mi a feladata a **hálózati kártyának és a modemnek**?

A PC-ket napjainkban nem csak önálló számítógépként, hanem helyi hálózatok (LAN) munkaállomásaként is hasznosítják. A helyi hálózathoz való kapcsolódás illesztő egysége a hálózati kártya, amely vagy a számítógépházban az alaplaphoz csatlakozó kártya, vagy az alaplapi vezérlő-áramkörkészletnek (chipset) a része. Ha a PC telefonvonalon kapcsolódik az Internethez, akkor az ehhez szükséges digitális-analóg konverziókat és hibajavítást végző modem is a konfiguráció része.

11. Milyen **részegységeket** tartalmaz a **számítógépház**?

A számítógépházat felnyitva, ezen belül a központi egység funkcionális részegységeit találhatjuk meg: a processzort, a sínrendszert, a memóriát, a háttértároló egységeket (merevlemez, optikai tároló egység) és az interfészeket.

12. Mi a feladata a PC **interfészeinek**?

Mivel az egyes perifériák és háttértárolók sebessége és működése igen eltérő, ezért az I/O eszközök és a sínrendszer közé olyan interfészeszközöket (csatoló, adapter, vezérlőegység, periféria sín) kell beiktatni, amelyek ezek együttműködését és összekapcsolhatóságát biztosítják.

13. Mi a feladata az **alaplapnak**?

A PC központi részegységei közül a processzort, a sínrendszert, a rendszermemóriát (RAM) és néhány periféria-interfész áramkört egy nagy nyomtatott áramköri kártya fogja össze, amit a számítógép alaplapjának (mainboard) nevezünk.

Az alaplap

1. Mit nevezünk **alaplapnak**?

Központi szerepet betöltő, a részegységeket és a szabványos periféria csatlakozásokat összefogó, köztük az adatáramlást biztosító nagy áramköri kártya.

2. A PC **milyen jellemző tulajdonságait** határozza meg az alaplap?

1. a PC fizikai felépítését
2. a számítógéprendszer teljesítményét
3. a számítógép konfigurációjának kialakítását, annak továbbfejleszthetőségét

3. Melyek az alaplap fő **funkcionális részegységei**?

1. processzorfoglalat
2. memóriefoglalatok
3. vezérlő áramkörkészlet (chipset)
4. sínrendszer és a kártyahelyek
5. perifériák csatlakoztatási helyei

4. Mi a **processzorfoglalat** feladata?

A processzorfoglalat egy szabványos csatlakozóhely, mely biztosítja a processzor mechanikai és elektromos összekapcsolását az alaplappal. A foglalat meghatározza a processzor kivezetéseinek számát és a feszültségszintet.

5. Milyen áramköröket tartalmaz a **chipset**?

Az alaplapok áramkörkészlete (chipset vagy lapkakészlet) tartalmazza a vezérlő és interfészáramköröket a sínrendszerek, a memória, az I/O rendszer, és az alaplapra integrált perifériák, háttértároló-vezérlések, portok és csatolók számára.

6. Hol találhatóak a **PCIe v2.0, illetve v3.0 vezérlőáramkörei**?

Az alaplap PCI Express (2.0 vagy 3.0 verzió) x16 kártyahelyében.

7. Mit jelent a **PnP**?

A PnP egy szabványcsoport, amely lehetővé teszi, hogy egy új I/O eszköz üzembeállítása esetén a periféria használatához szükséges beállításokat a számítógéprendszer automatikusan, emberi beavatkozás nélkül elvégezze.

8. Mi a hardver működési felügyelet (**monitoring**) feladata?

A hardware monitoring vagy működési felügyelet biztosítja, hogy az alaplap automatikusan és folyamatosan figyelemmel kísérje saját működését, időben jelezze a felhasználó számára a hardver meghibásodásait (például túlmelegedés).

9. Mit határoz meg az **APM** szabvány?

A fejlett energiagazdálkodás bevezetése (APM = Advanced Power Management szabvány 1991) biztosította a rendszer és a hardver részek energiellátásának szoftverrel történő vezérlését, és az eszközök használatának átmeneti szüneteltetése esetén ezek részbeni, vagy teljes kikapcsolását a tápellátásból.

10. Mi a különbség a **soft off** és a **stand by** állapotok között?

A soft off a számítógép logikailag kikapcsolt állapota, ilyenkor csak az alaplap néhány áramköre marad aktív állapotban.

A stand by készenléti állapotban a PC nagy energiafogyasztó részek energiájukat elvesztik, hogy a rendszermemória tartalma elveszne. Emiatt a számítógépet az ismételt használat előtt nem kell újraindítani. Ebben az üzemmódban kikapcsol a monitor, leáll a merevlemez motorja, csökken az órajel, leállnak a ventilátorok (a tápegység kivételével).

11. Milyen szabvány vonatkozik a hardver **komplex menedzselésére**?

A hardver komplex menedzselését az APCI (Advanced Configuration and Power Interface = Fejlett Konfigurációs és Energiagazdálkodási Interfész) nyílt, ipari szabvány 1996-ban történő megalkotása tette lehetővé.

12. Mi az **OSPM**?

Operating System-directed configuration and Power Management

13. Jellemezze az **ATX** szabványt!

Az AT (Advanced Technology) ipari szabvány eXtended változata, jellemző rá a processzor és tápegység közös hűtése, valamint a könnyen hozzáférhető memória foglalatok.

14. Mi a **tápegység** feladata?

A számítógép részegységei speciális egyenáramot igényelnek, ezért az asztali PC működtetéséhez a váltóáramú elektromos hálózat 230 V-os, 50 Hz-es feszültségét át kell alakítani kisebb feszültségű egyenárammá. Ezt a feladatot végzi el a számítógépházban található tápegység.

15. Milyen részekből áll az **alaplap specifikáció** összefoglaló táblázata?

Jó áttekintést ad a legfontosabb műszaki jellemzőkről, melyek legtöbbször elegendőek például egy konfiguráció módosítás megtervezéséhez.

A firmware interfész: BIOS, UEFI

1. Mit nevezünk **BIOS-nak**?

A BIOS az angol Basic Input / Output System kifejezés rövidítése, ami magyarul alapvető bemeneti/kimeneti rendszert jelent. Ez a hardvereszközök kezeléséhez szükséges alaprutinokat tartalmazza, ideértve a számítógép bekapcsolásakor végzett öntesztet (POST program), a konfigurációs paraméterek kezelését (SETUP program) és az operációs rendszer teljes körű betöltésének elindítását (BOOT program) végző szoftvert is.

2. Milyen **eszközkezelő** részei vannak az **operációs rendszernek**?

1. Logikai eszközkezelő rutinok
2. Hardverszintű eszközspecifikus rutinok (driver, bios)

3. Milyen **részekből** áll a **rendszer BIOS** (alaplapi flash BIOS)?

Ez tartalmazza a Setup, POST, Boot programokat és a PCI autókonzfigurációs rutinokat, a LAN és PnP támogatáshoz szükséges programokat.

4. Hogyan lehet aktualizálni az **eszközspecifikus BIOS-t**?

Célszerű az egyes hardvereszközökhöz tartozó BIOS rutinokat és adatállományait – mivel ezek változatlan formában működnek, amíg az eszközt használjuk – fizikailag az adott eszköz egy nem felejtő tárolójában elhelyezni. Így az eszköz cseréjekor (alaplapcsere, új grafikus kártya, vagy SCSI, SAS vezérlő stb.) a hozzátartozó BIOS aktualizálása is automatikusan biztosított.

5. Mi célt szolgál a **CMOS-RAM**?

Fontosabb BIOS paraméterek tárolására, és a számítógép kikapcsolása esetén is folyamatosan vezetni kell a dátum, óra, perc, másodperc értékét.

6. Milyen **adatokat** tartalmaz a **CMOS-RAM**?

1. másodperc, perc, óra
2. év, hó, nap
3. boot eszköz
4. merevlemez paraméterei
5. rendszermemória jellemzői
6. sínrendszer és alaplapi vezérlőáramkör-beállítások
7. energiagazdálkodás beállításai
8. BIOS jelszó (általában átalakított formában)

7. Mi a feladata a **POST** programnak?

A POST program egy hardverkomponens esetén (példánkban a monitorvezérlő kártya)

teszteli az adott egység működését (felhasználva a hardverkomponens BIOS-át is, ha van), majd felépíti a memóriában a hardverkomponens I/O rutinjait tartalmazó eszközközkezelő programrészeket.

8. Hogyan történik a **bootolás**?

- A processzor vizsgálata;
- A CMOS-ra és a BIOS-ra tesztösszeg előállítása, és összehasonlítása a letároltakkal;
- A DMA vezérlő tesztelése és üzembeállítása;
- A hagyományos USB és a billentyűzet tesztelése és működésbe állítása;
- A rendszermemória első 64 Kbájtos részének tesztelése;
- A megszakítási rendszer tesztelése, és üzembe állítása;
- A cachevezérlő tesztelése és inicializálása;
- A monitorvezérlő kártya tesztelése és működésének beindítása (ha hibás, fűtőjelek);
- A memóriaméret meghatározása a CMOS adatokból, és a memória tesztelése;
- A CMOS-RAM-ban megadott boot eszköz tesztelése és inicializálása.

9. Mi a feladata a **SETUP** programnak?

A CMOS-ban eltárolt konfigurációs beállításokat a BIOS Setup programjával lehet megváltoztatni, melynek beállítási lehetőségeit az alaplap kézikönyve tartalmazza.

10. Hogyan **frissíthető** a BIOS?

A BIOS frissítése történhet a Web-ről, illetve egy merev vagy optikai lemezről, vagy USB Pendrive-ről.

11. Mi a **BIOS Run Time Service**?

Az operációs rendszer betöltését követően a BIOS a memóriában elhelyezkedő eszközközkezelő rutinyűjteményként működik, és a hardver vezérléséhez egy funkcióhalmazt biztosít az operációs rendszer és a felhasználói programok számára.

12. Hogyan lehet felhívni a **BIOS-rutinokat** programból?

Ezeknek az eszközközkezelő rutinoknak a memóriában való felépítését a BIOS POST programrésze végzi, amelyhez felhasználja az alaplapi (rendszer) BIOS-t, a felhasználó által a BIOS Setup programja futása során megadott, és a CMOS-ban tárolt adatokat, valamint az egyes hardvereszközökhöz tartozó speciális BIOS-okat.

13. Ismertesse a **PnP szabvány** célkitűzéseit!

1. teljes kompatibilitás a korábbi, nem PnP működésű hardverkomponensekkel
2. maximális támogatása a PnP hardvereszközöknek: azonosítása és konfigurálása a PnP működésre képes kártyáknak
3. a hardverkonfiguráció dinamikus kezelése szoftvereszközökkel
4. a hardveresemények (pl. egy bővítőkártya kicserélése) felismerése és kezelése

14. Mi az **SMBIOS**?

Az SMBIOS szabvány, mely egy olyan adatstruktúrát határoz meg, melyben adatokat tárolhatunk a számítógéprendszeréről és annak részegységeiről. Segítségével a rendszeradminisztrátor információkat kaphat a számítógéprendszer típusáról, működési állapotáról, az egyes rendszerkomponensekről stb.

15. Mi az **UEFI**?

Az EFI továbbfejlesztésére 2005-ben megalakult az Unified Extensible Firmware Interface elnevezésű fórum, mely célul tűzte ki a BIOS fokozatos kiváltását. Ehhez -az X86-os kompatibilitás miatt- egy szabványos csatolófelületet definiáltak az operációs rendszer és a használt hardver platform firmware-je közé (ez lehetővé teszi, hogy az UEFI alatt PC BIOS, vagy Open Firmware rutinok egyaránt futóképesek legyenek).

16. Adja meg az **UEFI** legfontosabb **jellemzőit**!

1. Az UEFI gyakorlatilag az összes korszerű processzorral kompatibilis (processzor független eszközmeghajtó környezet)
2. Az UEFI része az ACPI és az SMBIOS
3. Az UEFI szolgáltatásai átfogják boot, és a run-time szervizt (pl. idő)
4. A felhasználó grafikus felületen állíthatja be a Setup paramétereket.

Az alaplapi vezérlőáramkör készlet

1. Milyen **áramköröket** tartalmaz a **chipset**?

Olyan áramkörcsoportokat tartalmaz, amelyek a PC fő hardveregységeinek, azaz a memóriáknak, a buszoknak, az I/O eszköz interfészeknek a vezérlését végzik.

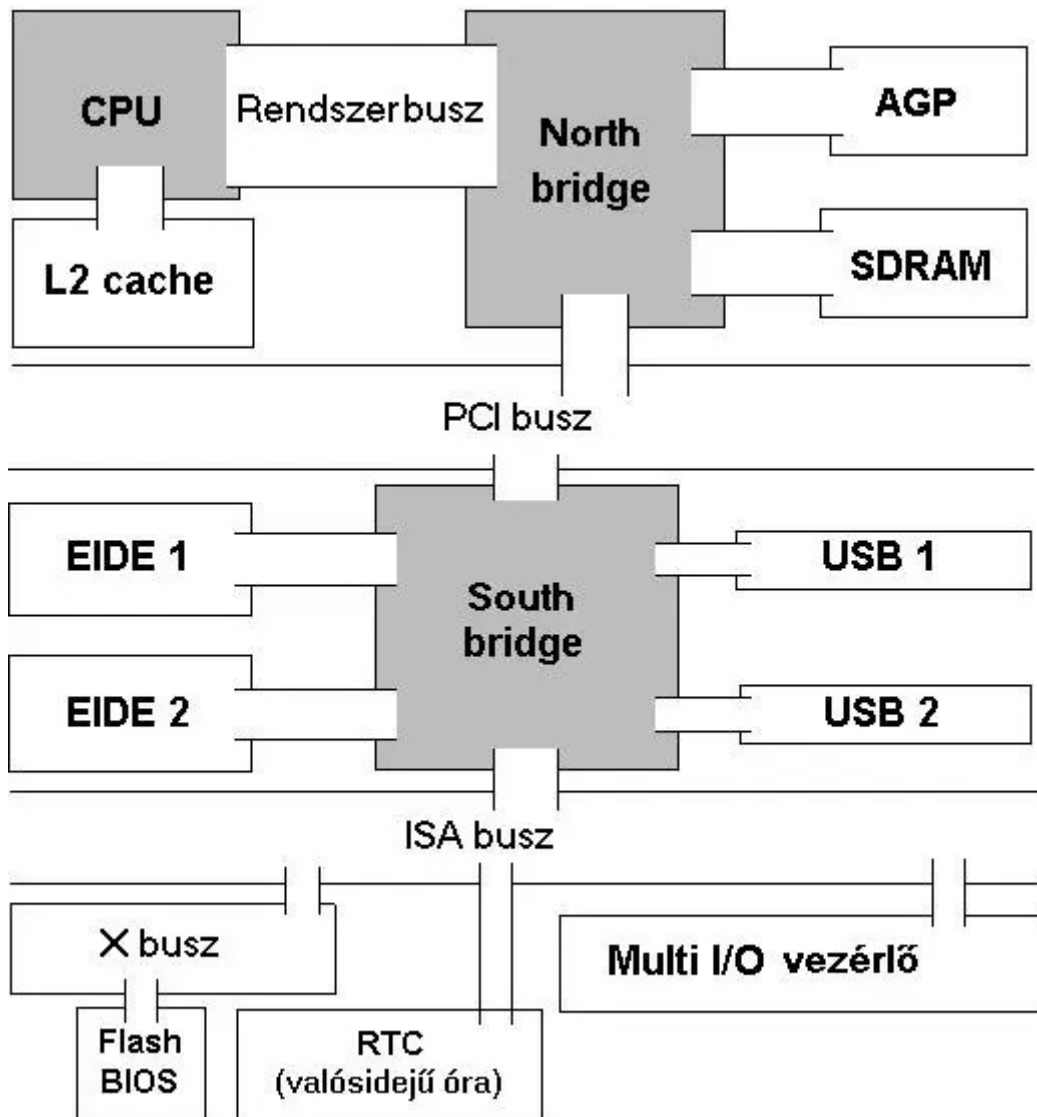
2. Mi a **memóriavezérlő áramkörök** feladata?

1. a másodlagos gyorsító tár (L2 cache) vezérlése
2. a rendszerbusz vezérlése
3. a rendszermemória hozzáféréseinek és frissítésének vezérlése
4. a grafikus busz (AGP) vezérlése
5. PCI hostbridge funkció ellátása

3. Mi az **I/O vezérlő áramkörök** feladata?

- az I/O buszok vezérlése
- merevlemez-csatoló kezelése
- a CMOS kezelése
- a valós idejű óra áramkör (RTC) kezelése
- a megszakítás-vezérlés
- a DMA-vezérlés
- az USB vezérlése
- PCI-ISA bridge funkciók ellátása
- a Multi I/O vezérlés, amely biztosítja meghatározott perifériák: billentyűzet, PS/2
- soros (COM) és párhuzamos port (LPT) és floppy meghajtó vezérlését

4. Milyen részekből épül fel a **bridge** architektúrájú **chipset**, és a részeknek mi a feladata?



Ekkor a memóriavezérlő áramköröket északi (North Bridge), az I/O vezérlő áramköröket pedig déli hídnak (South Bridge) nevezik.

5. Miben különbözik a **hub** és a **bridge** architektúrájú chipset?

1. a korábban a monitorvezérlő és 2D, 3D gyorsítókártyák által végzett feladatokat egy grafikus processzor végzi, melyet a GMCH-ba (Graphics and Memory Controller Hub) integrálták
2. az FWH hub egyrészt a Flash-memóriában tárolt BIOS kezelését biztosítja, másrészt egy fizikai véletlen számgenerátort tartalmaz, amely például titkosításra (rejtjelezés) használható
3. az AC97 CODEC interfész egy olcsóbb lehetőséget kínál a hangkártya és a modem helyett. Ezen keresztül kapcsolódhat az alaplaphoz az AMR (Audio Modem Riser) kártya, amely lényegében csak a telefonvonal kezeléséhez szükséges áramköröket tartalmazza. Természetesen ekkor a hangkártya társprocesszora által végzett feladatokat a CPU-nak kell átvállalnia
4. a 810-es áramkörkészlet nem támogatja már az ISA buszt, így az ISA-PCI bridgeáramkörök megtakaríthatók. Ennek viszont az az ára, hogy a 810-es áramkörkészlettel ellátott alaplapokat tartalmazó PC konfigurációban az ISA eszközök már nem alkalmazhatók

6. Mit jelent az **északi híd betolása** a processzorba és miért került erre sor?

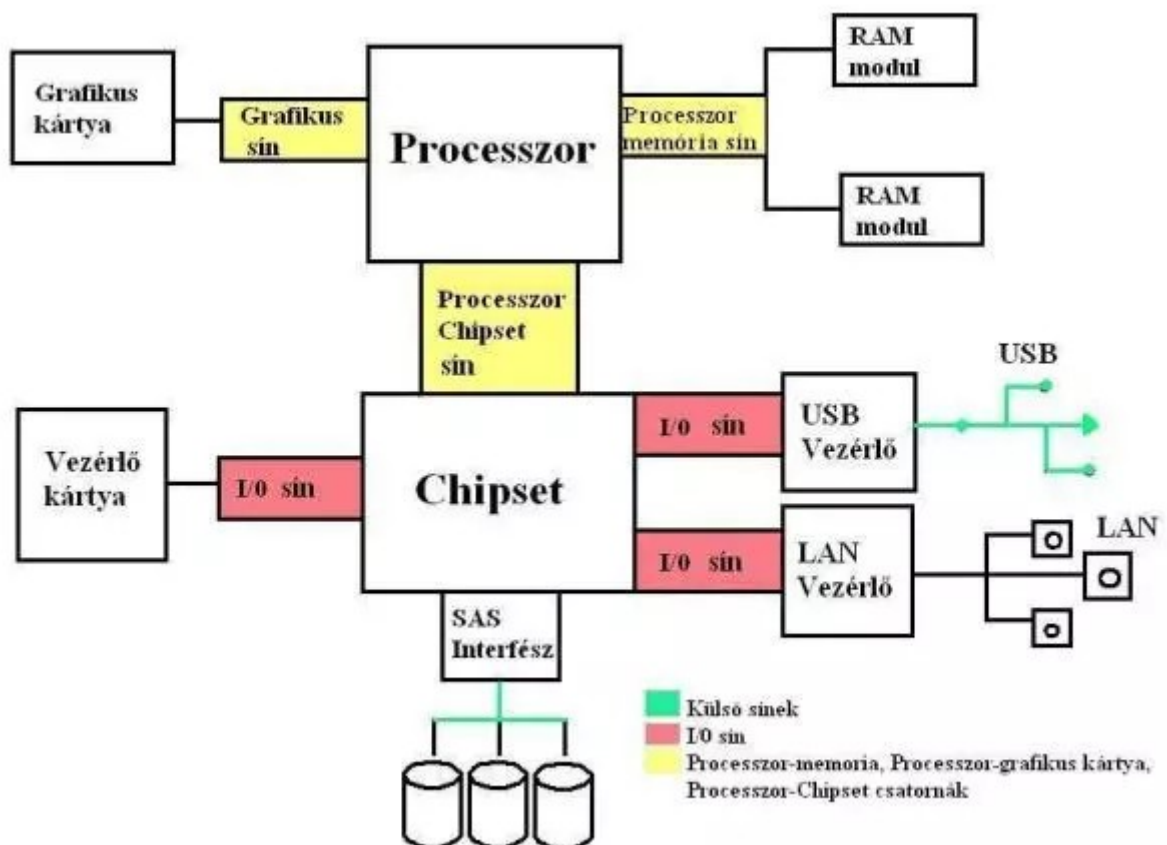
A 2000-es évek végére a PC teljesítményének növelésében a processzor és a memória, és grafikus alrendszer vezérlése közötti adatátviteli csatorna kapacitása jelentette a szűk keresztmetszet. A megoldást erre a chipset memória, grafikus és sín vezérlő áramköreinek a processzorba való integrálása jelentette.

7. Milyen **vezérléseket** tartalmaz a **PCH**?

1. PCIe ROOT
2. PCI Bridge
3. USB Host Controller
4. LAN vezérlő (Gbit Ethernet)
5. Audio alrendszer
6. SATA (SAS, RAID stb.) csatolók
7. IEEE 1394 vezérlő
8. Real-time óra, CMOS
9. Megszakítás vezérlő

A PC sínrendszerei

1. Milyen részekből áll a **PC kommunikációs architektúrája**?



2. Milyen **I/O buszok** lehetnek a **PC-ben**?

1. ISA busz, amely régi, lassú és ma már elavult sínrendszer
2. PCI busz, amely a Pentium gépcsaládba tartozó számítógépek 1990-es évekbeli általánosan használt I/O buszrendszere
3. PCI Express busz, amely a PCI buszprotokollt felhasználva, egy nagy teljesítményű, soros pont-pont összeköttetéssel valósítja meg a 2003-tól az adatátvitelt az I/O eszközevezérlésekhez, csatolókhöz

3. Hogyan működik a **PCI Bridge**?

Két PCI egység a PCI Bridge-n keresztül adatot cserélhet, ha közben olyan program fut, amelynél processzor nem címzett meg egy PCI egységet. Így a rendszermemória és egy I/O eszközevezérlő, vagy két I/O eszközevezérlő processzortól független adatcseréjét is biztosíthatja a PCI sín.

4. Jellemezze a **PCI busz arbitrációját**!

1. sínfoglalási kérelem jelzése = REQ (Request)
2. sínfoglalás engedélyezése = GNT (Grant)

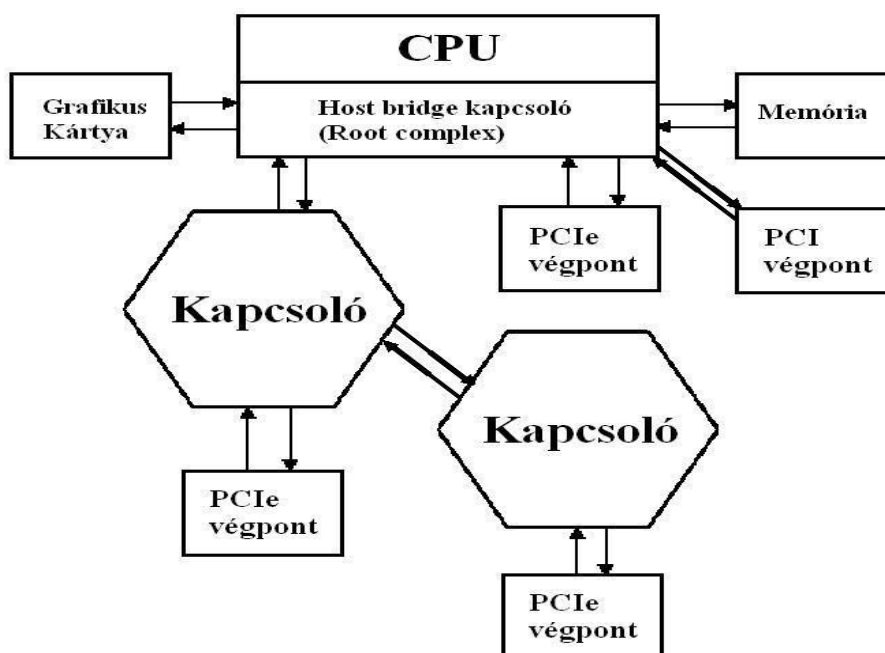
5. Milyen követelmények szerint fejlesztették ki a **PCI Express sít**?

1. a sínrendszer-architektúrát (ahol az eszközök több, párhuzamosan működő közös adatátviteli vonalat használnak) a kommunikáló eszközök közötti olyan pont-pont összeköttetésekkel kell felváltani
2. nagyteljesítményű soros adatátvitellel kell megoldani a kommunikációs feladatot;
3. az új I/O rendszernek az I/O busz kiváltása mellett a hálózati csomópontokkal, más processzorokkal, tároló alrendszerekkel való kommunikációs kapcsolatokat is biztosítani kell
4. a kommunikációs vonalak fizikai megvalósítása tegye lehetővé az üvegszálak összeköttetést
5. az adatátviteli összteljesítménye egy I/O kapcsolatrendszernek (Switch) meg kell, hogy haladja a 10 Gbit/sec-et, a hozzátartozó adatutak (lane) teljesítménye pedig legalább 2,5 Gbit/sec legyen

6. Mi a **PCIe lane és link**?

A PCIe sínrendszerben a soros pont-pont összeköttetés alapeleme fizikai szinten két-két vezeték párból épül fel, amelyek közül az egyik adatküldésre, a másik adatfogadásra szolgál. Ezt az egyidejűleg kétirányú adatátvitelre is képes duplex összeköttetést adatútnak (angolul lane=sáv) nevezik. Az adatuton történő adatátvitel logikai alapegysége a 8 bites adatcsomag, így egy lane tulajdonképpen bájt sorozatokat visz át (full duplex byte stream).

7. Mutassa be a **PCIe sín kommunikációs architektúráját**!



8. Hogyan biztosítják a **PCI és PCIe sínék kompatibilitását**?

A PCI Express szabványban alkalmazott hálózati rétegmodell biztosítja, hogy a különböző adatátviteli teljesítménnyel működő linkek a felső rétegekben (viszonylati, megjelenítési és alkalmazási rétegek) a PCI busz protokollal kompatibilisek legyenek.

9. Jellemezze az **FSB** processzor-memória sít!

A Frontside Bus fizikailag a memóriát és az alaplapi vezérlőáramkör-készletet köti össze a processzorral. Ebből fakadóan ezek az eszközök az FSB órajele által meghatározott sebességgel működnek.

10. Jellemezze az **DMI** processzor-memória sít!

A DMI (Direct Media Interface) egy interfész szabvány és egy összeköttetés az alaplapon, mely az alaplapi áramkörkészlet (chipset) északi és déli hídját kapcsolja össze.

11. Jellemezze az **QPI** processzor-memória sít!

A QPI (Quick Path Interconnect) egy nagyteljesítményű, a PCIe soros adatátvitel elvén működő, dedikált processzor összeköttetés.

12. Hogyan működik a rendszersín **Burst-Modus-ban**?

Ha a rendszersín gyorsított átviteli módban (Burst-Modus) működik, akkor legfontosabb feladata a processzorba épített L1 (on-chip) utasítás- és adatcache sorainak feltöltése az L2 cache-ből illetve a rendszermemóriából (cache-Line-Fills), valamint az L1 cacheben megváltoztatott adatok visszaírása a memóriába (write back).

13. Hogyan működik a rendszersín **egyes átviteli módban**?

Az I/O címtartományba és a rendszermemória cachetárral nem gyorsítható részébe (például a frame buffer esetén) eső címek esetében a Pentium rendszerbusza egyes átviteli módban működik. Ekkor a címátvitelt követően a sín az adatokat 64 bit = 8 bájtos egységekben továbbítja. A cache vezérlőnek egy vezérlőjel magas szintje jelzi, hogy nem kell az átvitt adatblokkal a cache egy sorát feltölteni, ennek hatására az adatok egy regiszterbe kerülnek. Egyes átviteli módban a memóriacímeknek 8-al, burst üzemmódban pedig 32-vel oszthatónak kell lenniük. Természetesen a 8 bájtos egységek (quadword) mellett a processzor a memória egyedi bájtjaihoz, szavaihoz és dupla szavaihoz is hozzáférhet. Erre a célra a BE0, BE1, ... BE7 bájt kiválasztó vezérlő jeleket alkalmazzák.

14. Milyen **adatútakat (lane)** tartalmaz a **DMI**?

A DMI legtöbb jellemzője megegyezik a PCIe sínével: többszörös adatutak (lane), pont-pont nagyteljesítményű soros összeköttetés. A legtöbb realizációban a DMI 4x linket használ, de néhány hordozható gépben előfordul a 2x kiépítés is. A 4x linket használva a DMI az 1.0 verzióban 10 Gbit/sec, a 2.0 verzióban, pedig 20 Gbit/sec adatátviteli teljesítményt biztosít.

15. Milyen **adatútakat (lane)** tartalmaz a **QPI**?

Minden QPI alapegység két 20 adatutas (lane) pont-pont összeköttetésből épül fel, mely 2 órajel alatt 80 bit adatmennyiséget (fits) visz át mindkét irányba (64 bit adat, 8 bit a hibafelismeréshez, 8 vezérlő bit a hírfejhez). Ezeket a 8 bájtos adataegységeket a protokoll szintjén adatcsomagokká kapcsolják össze, melyek mérete általában a cache sorok méretével egyezik meg. Egy ilyen QPI alap-

összeköttetés teljesítménye 25, 6 GB/sec.

16. Miért jött létre a PC-ben a grafikus busz?

Az 1990-es évek második felében a nagyfelbontású monitorok, a 3D grafika (különösen a folyamatos mozgóképeket előállító, egyre inkább fotorealisztikushoz közelítő képet generáló játékprogramok) miatt a monitoron megjelenített kép felépítéséhez évről-évre egyre több adatot kellett továbbítani az I/O buszon a monitorvezérlő kártyához.

17. Mi az AGP?

Az Intel a Pentium II processzorcsaládhoz vezette be a gyorsított grafikus portra vonatkozó sinszabványt, az AGP-t (AGP = Accelerated Graphics Port), amely a PCI sín protokollt jórészt változatlanul felhasználva, közvetlen (pont-pont), nagy teljesítményű adatátviteli kapcsolatot létesített a processzor, a rendszermemória (Front-Side Bus) és a monitorvezérlő kártya között.

18. Milyen rendszerelemeket kapcsol össze a PCIe x16?

A PC grafikus sínrendszere PCIex16 összeköttetéssel kezdetben az alaplapi vezérlő áramkörkészlet (chipset) északi hidját (GMCH) kötötte össze a grafikus kártyával, majd a Nehalem mikroarchitektúrától a grafikus sín közvetlenül a processzort (és a processzorba integrált memóriavezérlőt) kapcsolja össze a grafikus kártyával.

Input/Output interfészek

1. Mi az interfészek feladata?

1. az I/O sínrendszer és perifériák sebessége közötti különbségek áthidalása az adatok átmeneti tárolásával (pufferregiszterek igénybevételével)
2. a perifériák működésére vonatkozó utasítások fogadása és átmeneti tárolása (parancsregiszterek által)
3. az I/O eszközök állapotáról a programok (processzor) informálása (állapotregisztereken keresztül)
4. az I/O eszközökkel való kapcsolattartáshoz szükséges kommunikációs protokoll megvalósítása
5. adatkonverzió az I/O sín és a periféria formátumok között

2. Mit tartalmaznak az I/O portok?

Az I/O eszköz interfészek a PC I/O sínrendszeréhez kapcsolódhatnak, fizikai megvalósításuk szerint lehetnek alaplagra integráltak (ekkor az interfész áramköröket a chipset tartalmazza), vagy az alaplapba helyezhető kártyák (például grafikus kártya), vagy periféria sínrendszerek (például USB).

3. Mi az USB?

Az USB (Universal Serial Bus= univerzális soros busz) szabvány egy soros működésű periféria buszrendszert specifikál, melyet eredetileg a relatíve kis teljesítményű perifériák (például egér, billentyűzet stb.) csatlakoztatására szántak.

4. Milyen perifériák csatlakoztathatók az USB-hez?

- billentyűzet
- mutató eszközök (egér)
- pendrive

- nyomtatók
- digitális kamera
- játék konzol
- külső lemez (eSATA V3-tól)

5. Adja meg az **USB architektúrális** felépítését!

Az USB sín a chipset I/O vezérlő áramkörei között található USB host adapterrel kapcsolódik az I/O buszhoz, ezzel a sínrendszerre kapcsolt eszközök soros adatátvitellel kommunikálnak.

6. Milyen az **USB hálózat felépítése** (topológia)?

Az USB hálózat felépítése tiered star topológia (emeletes csillag szerkezet), azaz az eszközök szinteken kerülnek elhelyezésre, és hub-okhoz csatlakoznak.

7. **Jellemezze** az USB-t!

1. 1,5, 60, 640 MB/sec adatátviteli sebesség (V1, V2, V3)
2. elvileg 127 USB-eszköz kiszolgálása egy szinten, max. 5 db hub
3. soros adatátvitel
4. az eszközök sorosan felfűzhetők
5. lehetővé teszi a Plug and Play perifériatelepítést
6. az eszközök tápellátása USB-kábelén keresztül lehetséges (kis áramfelvételt igénylő perifériák)

8. Mi a **Firewire**?

A Firewire sínrendszer első változatát az SCSI kiváltására hozták létre, majd a számítástechnika alkalmazásában egyre nagyobb szerephez jutó közepes teljesítményű adatátvitelt igénylő multimédiás, grafikus perifériák miatt fejlesztették tovább.

9. **Jellemezze** a Firewire-t!

1. Adatátviteli sebesség 50 MB/s (IEEE 1394a 2000), illetve 400 MB/s (IEEE 1394b 2002)
2. Soros átvitel
3. Max. 63 periféria láncba felfűzve (daisy-chain topológia)
4. Gigabit Ethernet csavart érpárral (IEEE 1394c 2006)

10. Jellemezze a **soros portot**!

A soros port az adatbitek egymás utáni (soros) átvitelét teszi lehetővé egy adatátviteli vonalon az RS232 szabványos kommunikációs protokoll szerint.

11. Jellemezze a **párhuzamos portot**!

A párhuzamos vagy nyomtató portot (LPT = Line Print Terminal) főként karakteres nyomtatók (CENTRONICS defacto szabvány) csatlakoztatására használták a 90-es években, napjainkban a nyomtató port szerepét már az USB tölti be.

12. Mi az **IDE** lényege?

Integrated Drive Electronics, ami azt jelenti, hogy a mágneslemez vezérléséhez szükséges adatutak lerövidítése érdekében a vezérlő elektronikát fizikailag a lemezegységgel egybeépítik. Ebben az esetben az alaplapra integrált IDE csatoló host adapterként működik, azaz mint adatpuffer, összeköti az I/O buszt a lemezvezérlő-egységgel.

13. Hogyan történik az **IDE lemezek** vezérlése?

A merevlemez-műveletek irányítását általában a lemezegységbe beépített mikroprocesszorral és a vezérlési információkat tartalmazó, csak olvasható tárolóval oldják meg. Ezáltal a lemezvezérlő a számítógép fő processzorától függetlenül képes irányítani a lemezműveleteket, ellátja az ezzel összefüggő szinkronizációs és adatkonverziós feladatokat, ellenőrzi a végrehajtott műveletek hibátlanságát.

14. Mi az **ATAPI**?

Kibővített IDE hozzáférési parancskészlettel (ATAPI = AT Attachment Packet Interface) lehetséges CD-ROM és mágnesszalag csatlakoztatása is.

15. Mit jelent a **PATA** és **SATA**?

PATA: Parallel ATA, SATA: Serial ATA

16. Milyen **tulajdonságai** vannak a **SATA** csatolónak?

1. csökkentett vezetékcszámú kábel
2. külön pont-pont kapcsolat minden meghajtóval (nincs buszmegosztás)
3. gyorsabb adatátvitel (az első verzióban 150 MB/s, majd V2 300, V3 600 MB/s)
4. alacsonyabb amplitúdójú jelek (250 mV, 5 V-tal szemben)
5. hibaellenőrzés és hibajavítás képessége (32 bites CRC-kód alkalmazásával)
6. eszköz csatlakoztatása/eltávolítása a gép bekapcsolt állapotában is (Hot swapping)
7. NCQ (Nativ Command Queuing) optimalizálja a lemez hozzáférések sorrendjét minimalizálva a lemez olvasófejének mozgását

17. Adja meg az egyes **SATA verziók sávszélességét!**

1. SATA V1.0 1.5 Gbit/s (2003)
2. SATA V2.0 3 Gbit/s (2005)
3. SATA V3.0 6 Gbit/s (2008)
4. SATA V3.2 (SATA Express = PCIe sín SATA szoftver protokollal) 8 Gbit/s-16 Gbit/s (2012)

18. Mi az **eSATA**?

Külső SATA (eSATA = External SATA) alkalmazásához 2004-ben szabványosították a külső tárolók különböző kábeleit, a csatlakozókat, a jelszinteket, és alkották meg az eSATA specifikációját. Az eSATA az USB és a Firewire versenytársa, esetében a max kábelhossz 2 méter, átviteli sebessége 300 MB/sec.

19. Mi a **SCSI**?

Small Computer System Interface, azaz kisseámítógép-rendszer interfész, egy olyan szabványos csatoló, amely önálló sínrendszerként működik. Fizikai megvalósításában lehet PCI kártya vagy alaplapra integrált.

20. **Jellemezze** a **SCSI** csatolót!

1. a SCSI intelligens csatoló, ha megkapja a kommunikációra vonatkozó adatokat, akkor azt önállóan intézi, nem terheli a processzort, ezáltal segíti a multitasking üzemmódot
2. a SCSI egy buszrendszer, amelyre különböző típusú, "SCSI eszközök" csatlakoztathatók, pl. merevlemezek, streamerek, nyomtatók, szkennerek stb.

3. a különböző SCSI szabványok szerint a buszra csatlakozó eszközök 8, 16 vagy 32 bit szélességben képesek kommunikálni
4. a SCSI csatolóval az ATA-nál gyorsabb lemezek is csatlakoztathatók a PC-hez
5. a SCSI csatolók és lemezek műszaki megbízhatósága általában nagyobb az ATA-nál, ennek megfelelően áruk is jóval több lehet

21. Hogyan történik a **kommunikáció** a **SCSI** sínen?

A SCSI buszra csatlakozó eszközök lehetnek adatátvitelt kezdeményező (Initiator vagy Master), illetve az adatátvitelben résztvevő (Target vagy Slave) eszközök. A SCSI sínhez csatlakozó host adapter eszköz mindig Initiator.

22. Mit jelent a **SMART** technológia?

A SMART (Self Monitoring Analysis Reporting Technology) technológia a lemezegységet folyamatosan ellenőrzi (például a fordulatszámot vagy a fej távolságát a lemeztől), és előre jelzi a hibát, mielőtt az az adatok megsérülését eredményezné.

23. Jellemezze a **SAS** csatolót!

A SAS (Serial Attached SCSI) szabvány 2004-ben egy olyan pont-pont, soros kommunikációs protokollt specifikál, mely lényegesen gyorsabb adatkapcsolatot valósít meg, mint a korábbi SCSI interfész, ugyanakkor a kommunikáció továbbra is SCSI vezérlőparancsokkal történik. További előnye a SATA meghajtókkal való kompatibilitás, azaz SATA merevlemezek is csatlakoztathatók a SAS csatolókhöz (a SAS „ismeri” a SATA parancsokat), ami szerverkonfigurációk esetén lényeges lehet.

24. Adja meg az egyes **SAS** verziók **sávszélességét**!

- SAS-1 300MB/sec (2004)
- SAS-2 600MB/sec (2008)
- SAS-3 1200MB/sec (2012)

A PC processzorai

1. Milyen processzor működött az **IBM PC XT**-ben?

Az első IBM PC-ben az Intel 8086-os vagy 8088-as típusú processzorait építették be 1978. illetve 1979.-ben.

2. Melyik volt az első **ILP** processzor?

A 80286-os processzor első változata 1982-ben jelent meg.

3. Jellemezze az első **32 bites** processzort!

Az Intel első 32 bites processzora a 80386DX, 1985-ben jelent meg, 275 000 tranzisztort tartalmazott. Az első széria 16 MHz órajel frekvenciával rendelkezett, az ezt követő különböző változatok 20, 25, 33 és 40 MHz órajellel működtek. A belső regiszterek valamint az adat- és címsínek is 32 bitesek, így a processzor 4 Gb-át memória fizikai címzésére volt képes. Ez volt az első Intel mikroprocesszor, amely már alkalmazta a csővonal (pipeline) utasítás végrehajtást.

4. Jellemezze az első szuperskalár architektúrájú processzort!

Az első szuperskalár architektúrájú processzor az Intel által gyártott Pentium volt. Ez 1993-ban jelent meg, az első változatok 3,1 millió tranzisztort tartalmaztak, és bipoláris CMOS technológiával készültek, valamint 60 MHz órajellel működtek. A Pentium az első olyan processzor, amely 64 bites adatsínt használ, de továbbra is 32 bites címsínnel rendelkezik. További újdonság, hogy a külső órajel 50, 60 majd 66 MHz-re emelkedett, így (az órajel sokszorozásnak is köszönhetően) a Pentium processzorok 60 és 200 MHz órajel közötti sebességgel működtek.

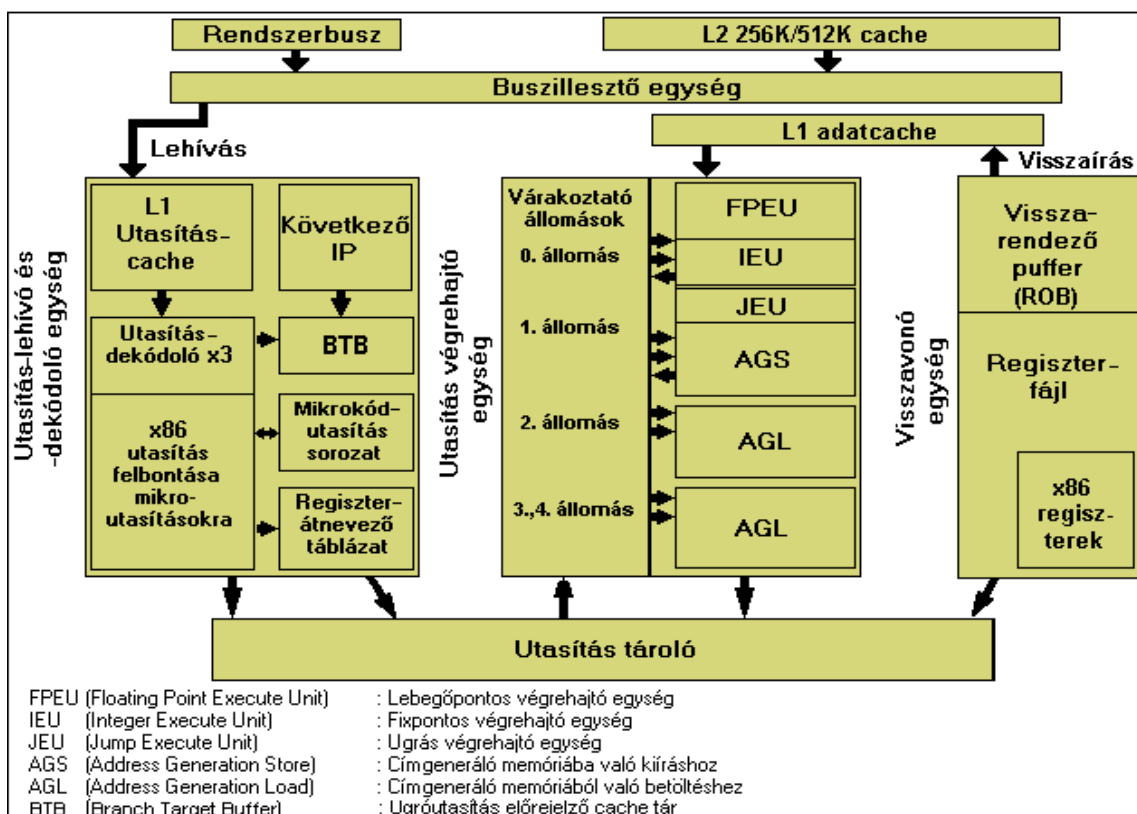
5. Mi jellemzi a Pentium MMX processzort?

Az Intel következő processzora a Pentium MMX volt, melyet a multimédiás alkalmazások hatékonyabb futtatása érdekében fejlesztették ki, és 1997 januárjában bocsátotta ki az Intel.

A Pentium MMX processzorok külső órajele 66 MHz volt, a különböző típusok 166-233 MHz közötti sebességgel működtek. A legnagyobb eltérés a Pentium és a Pentium MMX között a kibővített, multimédiás utasításkészlet. Ezért az olyan alkalmazások futtatása során, amelyek képesek kezelni az MMX utasításkészletet, nagy sebességekülönbséget tapasztalhatunk. Emellett a Pentiumhoz képest még további változtatások is történtek:

1. Megnövelt elsődleges cache: külön választott, 16 Kb-át méretű adat és 16 Kb-át méretű utasítás cache.
2. Mélyebb csővonal: a csővonal állapotok számát ötről hatra emelték.
3. Új MMX egység (8 db MMX regiszter, de ezek fizikailag megegyeznek a régi lebegőpontos regiszterekkel).
4. Továbbfejlesztett ugrás-előrejelzés 512 elemű BTB-vel. RSB = Return Stack Buffer, azaz visszatérési verem tároló a szubrutin visszatérési címek előrejelzésére.

6. Mutassa be a P6 processzormagos architektúrát!



7. Mi a **3DNow!**?

Az AMD 1998 nyarán mutatta be K6-2 3DNow! nevű utasításkészlet-bővítéssel ellátott processzorát. A processzor nagyon jó tesztteredményeket ért el a 3D grafikus alkalmazásoknál. Ezt a 21 darab új SIMD utasítás (a 3D-s alkalmazások által leggyakrabban használt műveletek) tette lehetővé, amelyeket a processzor vektorszámítógépként hajtott végre.

8. Mi az **SSE**?

Az SSE (Streaming SIMD Extension) utasításkészlet-bővítés lehetővé teszi, hogy a processzor a 3D grafikus műveleteket vektorszámítógépként (négy vektorkomponensre párhuzamosan) hajtsa végre.

9. Mit értünk **ISA architektúra** alatt?

A processzorok utasításkészlet szintű architektúrája (Instruction Set Architecture = ISA) a processzorarchitektúrának a szoftver oldali képe, azaz a programok által „látható” része, melyet a processzor által feldolgozható adatok (natív data), utasítások, regiszterkészlet, címzési módok, a processzor megszakítás és az I/O kezelése jellemeznek. Ezt gyakran a processzor programozási modelljének is nevezik.

10. Mit jelent a **processzor mikroarchitektúrája**?

A processzor mikro-architektúrája a processzor működését a mikroutasítások végrehajtásának szintjén jellemzi. A mikroarchitektúra megadja a hardver felépítését és a részegységek együttműködését a funkcionális blokkok szintjén: a végrehajtó egységek (ALU, FPU LOAD/STORE egységek) száma, típusa, a pipeline szervezése, az elágazás előrejelzés stb. Ennek szokásos szemléltetése általában egy blokkvéskéma.

11. Jellemezze a **Netburst** mikroarchitektúrát!

Az első Netburst mikroarchitektúrájú Pentium 4 processzort Willamette néven az Intel 1,4 és 1,5 GHz órajel-frekvenciával 2000 novemberében dobta piacra. Az új hetedik generációs, 42 millió tranzisztort tartalmazó processzorral az Intel - még megmaradva a 32 bites keretek között - a 3D grafika (renderelés) és MPEG kódolás/dekódolás, a beszédfelismerés és rejtjelezés felgyorsítását tűzte ki célul.

12. Mit jelent az **IA-64**?

Az IA-64 (Intel Architecture 64) egy az Intel és a Hewlett-Packard által közösen kifejlesztett utasításkészlet-architektúra (instruction set architecture, ISA; az utasításkészlet-architektúra az utasításokon kívül a felépítés több más elemét is meghatározza, pl. az adatformátumokat, regisztereket, címzési módokat, memóiafelépítést, a megszakítás- és csapdakezelést, valamint a kapcsolatot a külvilág felé - I/O).

13. Mit jelent az **x86-64**?

Az x86-64 az x86-os architektúra 64 bites leszármazottja: az x86 utasításkészlet-architektúra 32 bites (IA-32) verziójának kiterjesztése.

14. Jellemezze az **AMD64** architektúrát!

Az AMD64 legfontosabb jellemzői a 64 bites általános célú regiszterek támogatása, 64 bites egészekkel történő aritmetikai és logikai műveletek elvégzése, és a 64 bites virtuális címek.

15. Mit jelent a [Hyper-Threading](#) technológia?

A legtöbb operációs rendszer a futtatandó programfolyamokat olyan független "szálakra" (threads) bontja fel, amelyekhez a processzorhasználat önállóan ütemezhető. Emellett számtalan fontos alkalmazásnál a feladat jellege is olyan, hogy a szálakra bontás viszonylag egyszerűen megoldható (adatbázis-motor, 3D-grafika raytracing renderelés a képernyő részekre bontásával stb.).

16. Mit értünk [többmagos processzor](#) alatt?

A két- vagy több- magú processzorokba a processzor végrehajtó egységeit fizikailag is meg többszörözik a chipen, az lásd alábbi ábrán.

17. Jellemezze a [Core](#) mikroarchitektúrát!

Az új Core mikroarchitektúra tervezésekor az Intel alapvető célja a processzor teljesítményének növelése mellett az energiatakarékos működés volt.

18. Melyek a [Core](#) mikroarchitektúrájú processzorok fontosabb [paraméterei](#)?

1. Két processzormag egy chipre integrálása, a négymagos változatoknál megduplázták a kétmagos chipet a tokózáson belül.
2. Megszüntették a Hyper Threading lehetőségét.
3. Nincs Hyper Pipeline, a pipeline fokozatok számát visszavették 14-re.
4. A fogyasztás csökkentése érdekében a chipnek nem használt részei (akár a teljes chip is) automatikusan lekapcsolódnak (clock gating)
5. Az L1 cache méretének növelése mellett (32KB utasítás és 32KB adatcache), a két mag egy chipen belül megosztva használja a közös L2 cache-t.

19. Milyen akadályozó problémákat kellett megoldani a [Nehalem](#) mikroarchitektúra tervezésekor?

1. A korábbi mikroarchitektúrák nem támogatták megfelelően a tervezés és gyártás moduláris megszervezését, a többféle teljesítményigényt lefedő, skálázható processzorok előállítását. (a 4 magos Core mikroarchitektúrájú processzorokat csak a chip duplázásával tudták előállítani)
2. A 4-8 magos processzorokkal ugrásszerűen növelt utasítás-feldolgozó kapacitást a rendszersín már nem tudta kiszolgálni, nem tudta ellátni a processzormagokat ütemesen utasításokkal és adatokkal.
3. A chipset északi hídjába integrált, egyre nagyobb teljesítményt igénylő 3D grafikus feldolgozásokat a processzor és grafikus vezérlő közötti kommunikációs csatorna szűk kapacitása egyre inkább visszafogta.

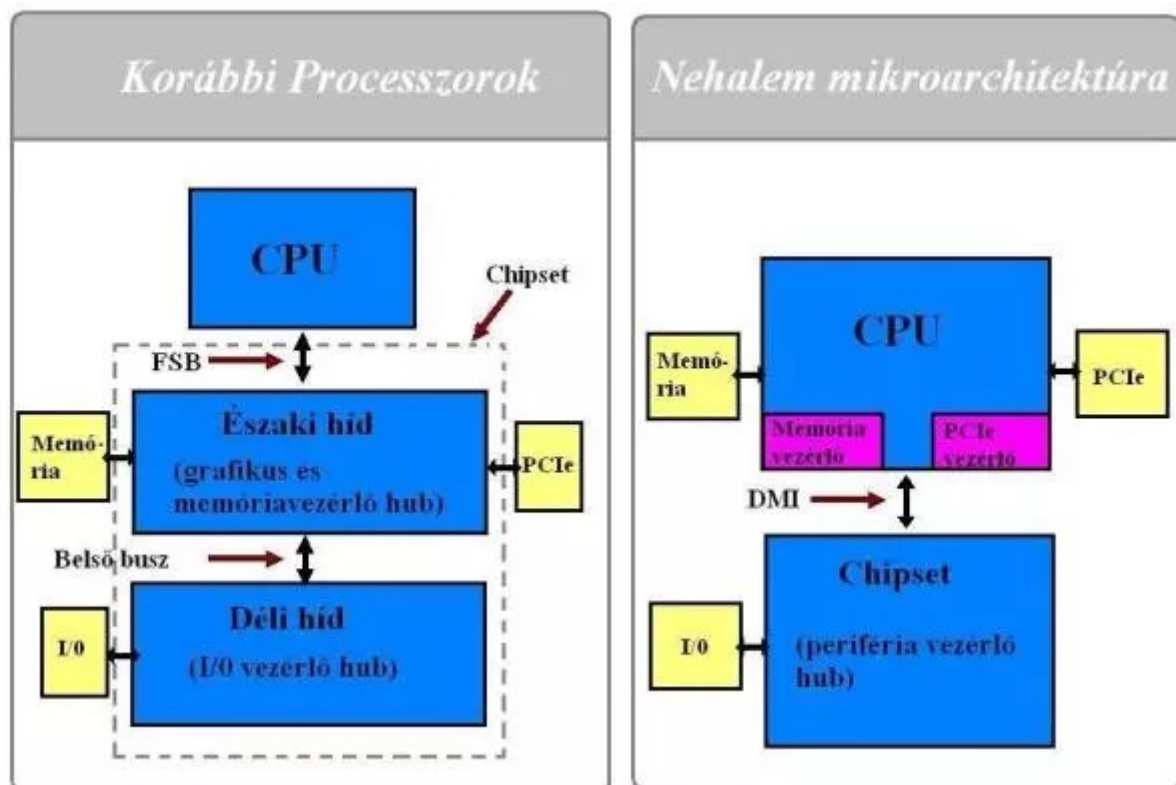
Ez alapján meg kellett oldani:

1. 8 magig (és a Hyper Threading visszahozásával 16 szálalig) egy chip-re integrálható, jól skálázható moduláris architektúra létrehozása, mellyel lehetővé válik a processzor egyes funkcionális elemeinek (magok, gyorsító tárhelyek, memóriavezérlő, kommunikációs-csatornák, stb.) külön-külön történő fejlesztése, és az elemekből többféle igényt kielégítő processzorok gyártása.
2. A chipset északi hídjának „betolása” a processzorba, azaz a memória, és a PCIe sínvezérlő integrálása a processzor chip-re.
3. A grafikus processzor integrálása a processzorba.

20. Melyek a Nehalem processzorok főbb **architektúrális** jellemzői?

1. Az architektúrából kiiktatták (az AMD Hyper Transport-hoz hasonlóan) a rendszerbuszt (FSB), a processzor a rendszer többi elemével nagy sávszélességű, pont-pont összeköttetésekkel (QPI, DMI) kommunikál.
2. A memóriavezérlőt, és a PCIe vezérlőt a processzorba integrálták. Az új IMC (Integrated Memory Controller) memóriavezérlő három nagy teljesítményű csatornája a DDR3-as memória modulokhoz jelentősen lecsökkenti a memória elérését.
3. A processzormagok ismét képesek két szál párhuzamos feldolgozására (Hyper Threading), ehhez megkettőzték a magok egyes erőforrásait (például regiszterfájl, RSB, TLB) illetve más erőforrásokat, kapacitásukat megnövelve felosztottak a szálak között (például a ROB mérete szálanként 64-re, összesen 128 bejegyzésre nőtt).
4. A Nehalem mikroarchitektúrájú processzorok az elődjeiknél lényegesen jobban skálázhatók. Natív módon, azaz az összes mag egy chipre integrálásával) előállíthatók 2,4, 6 ill. 8 magú változatok, magonként egy vagy kétszálú működéssel

21. Milyen **részegységeket** tartalmaz egy **Nehalem** processzor?



22. Hogyan **skálázhatók** a **Nehalem** mikroarchitektúrájú processzorok?

A Nehalem mikroarchitektúrájú processzorok az elődjeiknél lényegesen jobban skálázhatók. Natív módon, azaz az összes mag egy chipre integrálásával) előállíthatók 2,4, 6 ill. 8 magú változatok, magonként egy vagy kétszálú működéssel.

23. Milyen **végrehajtó futószalagok** működnek egy Nehalem mikroarchitektúrájú processzormagban?

Összesen 3 porton keresztül áramlanak át az adatok (Port 0/1/5) a három lebegőpontos, két egészszámos, illetve három SSE végrehajtó felé. E három porton

felül további három (Port 2/3/4) szolgál az utasításbeolvasásra vagy kiírásra (Load, Store Address, Store Data).

24. Milyen gyorsítótárak ([cache](#)) vannak egy Nehalem magos processzorban?

Az elsőszintű gyorsítótár a Nehalem-ben a Core-hoz hasonlóan 32-32 KB adat és utasításcache, melyek 8 és 4 utas csoportasszociatív táruk. A Nehalem másodszintű gyorsítótárának mérete 256 KB, és ez szintén 8 utas csoportasszociatív működésű. A harmadszintű, 2-12MB (a Westmere-től 4-30MB), 16 utas csoportasszociatív gyorsítótár megjelenése újdonság, L3 cache tárat eddig csak a szervergépek tartalmaztak. Az L3 cache megosztott a processzormagok között és inkluzív, azaz az összes L1 és L2 gyorsítótárban megtalálható információt tárolja.

25. Hogyan működik a [PCU](#)?

Ez a hardver egység felelős a magok, az északi híd és a harmadszintű gyorsítótár hőmérsékletének és áramfogyasztásának ellenőrzéséért, és szabályozásáért.

26. Mit értünk [Turbo Boost](#) technológia alatt?

A „Turbo Boost” technológia bekapcsolása esetén a PCU terheléstől és hőmérséklettől függően meg tudja változtatni az egyes processzormagok órajelét. Ha nincs szükség a processzor összes erőforrására, a PCU a nem használt magokat lekapcsolja, illetve a leterhelt magok órajelét a rendszer dinamikusan megemeli, amíg a processzor el nem ér egy adott hőtermelési értékét. A Turbo mód akár mind a négy aktív processzormag esetében is megemelheti az órajelet, amíg a PCU úgy érzékeli, hogy a hőmérsékletek rendben vannak és az aktuális fogyasztás a maximalizált érték alatt van.

27. Melyek a [Nehalem](#) architektúrájú processzorok fontosabb [paraméterei](#)?

Gyártás: 2008-2011

CPU órajel: 1.06-3.3 GHz

Magok száma: 2(2)-8(16) (szál)

Utasításkészlet: x86-64

Gyártástechnológia: 45-32 nm

1-4xQPI vagy DMI 4,8-6,4 GT/s

L1 64 KB (32+32) magonként

L2 cache magonként 256K

L2 szintű TLB és BTB

Processzorba integrált, közös használatú L3 2-12MB. (Westmere-től 4-30MB)

28. A Nehalem mikroarchitektúrájú processzorok tartalmazhatnak-e [GPU-t](#)?

Mivel ekkor már egyre komolyabb gondot okozott a processzoron kívüli grafikus vezérlő miatt a 3D grafikus feldolgozások lassúsága, a kétmagos Arrandale és Clarkdale processzorokat egy tokba integrált 32 és 45nm-es grafikus chippel gyártották.

29. Milyen célkitűzéseket valósított meg az Intel a [Sandy Bridge](#) mikroarchitektúra kifejlesztésével?

1. A korábbi többmagos processzorok működtetésével kapcsolatos tapasztalatokat felhasználva, egy kiértelt, jól skálázható, többmagos, egy chippel integrált processzor architektúra létrehozása.
2. A processzorba integrált GPU- val (Graphical Processor Unit) a 3D grafika és játékprogramok területén kiváltani a legkisebb teljesítményű (un. „belépő” szintű) professzionális grafikuskártyákat.

30. Melyek a Sandy Bridge processzorok fontosabb **architektúrális** jellemzői?

1. Egy chipre integrált, azonos gyártástechnológiával készített CPU, GPU, memória és PCI Express vezérlés.
2. Max. 8 fizikai és max. 16 logikai mag (Hyper Threading).
3. A processzorba integrált, megnövelt sebességű L3 cache, melyet a CPU-k és a GPU közösen használnak.
4. 256 bit/ciklus méretű, gyors gyűrű felépítésű (ring) belső csatorna biztosítja a kommunikációt a magok, a grafikus processzor, az L3 cache, a memória és a buszvezérlés között.
5. Megnövelt számú feldolgozó egységgel továbbfejlesztett grafikus processzor (GPU).
6. Új SIMD utasításkészlet: AVX = Advanced Vector Extensions 256 bites vektorutasításokkal.

31. Milyen **részegységekből** épül fel egy Sandy Bridge processzor?

- A „rendszerügynök” (System Agent) egység összefogja a processzornak a végrehajtó magokon, illetve a grafikus motoron kívüli részeit. Ide tartozik:
- Az IMC (Integrated Memory Controller) integrált memóriavezérlő (többcsatornás DDR3),
- A beágyazott display vezérlő,
- A PCIe vezérlő (PCIe x16),
- Az L3 cache vezérlő,
- A DMI vezérlő,
- A PCU vezérlés a Turbo Boost-al.
- A 4 db végrehajtó mag (az ábrán Core, kék színnel jelölve).
- A grafikus processzor (az ábrán világoszöld színnel jelölve).
- A L3 cache (Last Level Cache = LLC)
- A „körforgalmi adatbusz” (ring bus), amit az ábrán barna és piros szín jelöl.

32. Milyen különbségek vannak a **Nehalem** és a **Sandy Bridge** mikroarchitektúrájú magok működésében?

A Nehalem-mel összehasonlítva a Sandy Bridge processzormag utasításlelő és dekódoló része a kiegészült egy 1500 mikro-utasítás kapacitású cache tárral. (hasonlóan a Pentium 4 Execution Trace Cache-hez). Ha az utasításlelő egy olyan utasítást észlel, mely korábban már dekódolásra került és megtalálható a mikroutasítás cache-ben, akkor ennek az utasításnak a dekódolása kihagyásra kerül, a mikroutasításokat ez esetben a cache-ből veszi a processzor.

33. Mi az **AVX**?

Az új AVX (Advanced Vector Extensions) nevű vektor utasításkészlet a lebegőpontos utasítás feldolgozás gyorsítása érdekében egy gépi ciklus alatt 256-bites utasítások végrehajtását is lehetővé teszi. Az AVX utasításokhoz tartozó adatok feldolgozását 144 darab új 256 bites fizikai regiszter is segíti (YMM regiszterek). Az AVX utasítások a multimédiás, és a 3D grafikus feldolgozásokat gyorsíthatják, hatékonyabbá tehetik a tudományos és pénzügyi szimulációkat.

34. Milyen gyorsítótárak ([cache](#)) vannak egy Sandy Bridge magos processzorban?

1. 32-32KB L1 utasítás és adatcache magonként
2. 144 illetve 100 bejegyzéses L1 utasítás és adat TLB magonként
3. 256KB L2 cache és 512 bejegyzéses L2 TLB magonként
4. a CPU magok és a GPU által közösen használt L3 cache 1-20MB közötti kapacitással

35. Jellemezze a Sandy Bridge processzorok [GPU-ját](#)!

A Sandy Bridge mikroarchitektúrájú processzorok integrált GPU-ja a legkisebb teljesítményű professzionális grafikus kártyákkal kívánja felvenni a versenyt. Ezek 12 illetve 6 darab 128 bites vektorfeldolgozó egységet (shader) tartalmaznak, melyeket 24 elemű textúrázó blokk egészíti ki.

36. Miben különböznek a Sandy Bridge és az Ivy Bridge processzorok?

Az Ivy Bridge processzorok a korábbi Sandy Bridge mikroarchitektúrát lényegében változatlanul alkalmazzák, de a 22nm-es csíkszélességgel, 3D-s tranzisztorokra épülő gyártástechnológiával készülnek. A 3D-s tranzisztor alkotóelemei nem rétegszerűen helyezkednek el, hanem háromdimenziós struktúrát képeznek.

37. Mit jelent a 3D tranzisztoros [22nm-es](#) gyártástechnológia?

A térbeli felépítés előnye, hogy a 3D tranzisztorok kisebb alapterületűek, így jóval sűrűbben helyezhetők el egy szilíciumszeleten, a 32 nm gyártástechnológiához képest a 22nm 3D eljárással kétszeres tranzisztor-sűrűség érhető el.

38. Fogalmazza meg a processzorok fejlődésének fő irányai [2020-ig](#)!

Az Intel által nyilvánosságra hozott fejlesztési program alapján a következő években a következő mikroarchitektúrák megjelenése várható:

- Haswell 22nm, 2013
- Broadwell 14nm, 2015
- Skylake 14nm, 2015
- Skymont 10nm, 2017

Szakértők szerint 2020-ig még rátérhetünk 7 és 5nm-es gyártástechnológiára, de többen azt várják, hogy ez lesz az utolsó évtizede a szilíciumalapú technológiának.

A processzorok következő évekbeli fejlődésének várható jellemzői:

- Az integrált GPU teljesítménye tovább növekszik, a grafikus programcsomagok (Direct 3D, Open GL) funkcióit és videó kódolást teljeskörűen átveszi a hardver.
- Tovább folytatódik a vezérléseknek a processzorba történő integrálása (SOC = System-on-a-chip koncepció), a következő lépésben az alaplap PCH vezérlő áramkörei kerülhetnek a processzor chip-re.
- Folytatódik a párhuzamosítás a processzormagok többszörözésével (például Intel jóslat: 10 éven belül 48 CPU mag lesz a mobiltelefonokban), ehhez természetesen fel kell gyorsítani a párhuzamos szoftverek fejlesztését.