

Melléklet: Fogalomtár

1 bites összeadó

Két egybites szám összeadására képes áramkör, amely figyelembe veszi az előző helyiértékeken történt összeadás átvitelbitjét.

16 bites processzor

Olyan processzor, amely az adatokat jellemzően 16 bites egységekben dolgozza fel.

16 bites számítógép

Olyan számítógép, amely az adatokat jellemzően 16 bites egységekben dolgozza fel.

286-os PC

Az Intel 1982-es 80286 típusjelű 16 bites processzorára épülő személyi számítógép.

32 bites processzor

Olyan processzor, amely az adatokat jellemzően 32 bites egységekben dolgozza fel.

32 bites számítógép

Olyan számítógép, amely az adatokat jellemzően 32 bites egységekben dolgozza fel.

3D grafikus program

A vektorgrafika modellterében definiált objektumokat kezelő és raszteres képen megjelenítő program.

3D gyorsító társprocesszor

Önálló működésre alkalmas, ún. raszter processzor a monitorvezérlő kártyában, amely a vektorgrafikus adatokat konvertálja raszteres, azaz a képernyőn megjeleníthető formába.

3D gyorsítókártya

A vektorgrafikus modellterében meghatározott objektumok raszteres képen való megjelenítéséhez szükséges számításokat végző társprocesszorral (raszterprocesszor) és/vagy a modellterében az objektumok mozgatásához szükséges számításokat végző társprocesszorral (geometria processzor) ellátott monitorvezérlő kártya. (Régebbi architektúrákban a 3D gyorsítócsipet külön kártya tartalmazta.) A professzionális felhasználás (például CAD tervező rendszerek) mellett a játékprogramok lejátszásánál játszik meghatározó szerepet a 3D gyorsító kártya processzorának teljesítménye és a kártyán található memória mérete.

3D modelltér

L.: Vektorgrafikus modelltér

8 bites processzor

Olyan processzor, amely az adatokat jellemzően 8 bites egységekben dolgozza fel.

8 bites számítógép

Olyan számítógép, amely az adatokat jellemzően 8 bites egységekben dolgozza fel.

80286 = i80286

Az Intel 1982-es, 16 bites processzor típusjele.

80386 = i80386

Az Intel által 1985-től gyártott 32 bites processzor típusjele.

80486 = i80486

Az Intel által 1989-től gyártott 32 bites processzor típusjele.

8086-os emuláció

Az Intel és vele kompatibilis processzorok valós üzemmódja, amelyben a processzor úgy viselkedik, mintha a felhasználói program i8086-os processzoron futna (például 32 bites regisztereknek csak az alsó 8 bitjét engedni használni, az utasításkészlet is ennek megfelelően csökkentett).

810-es csipkészlet = i810-es csipkészlet, 810-es áramkörkészlet

Az Intel első hub architektúrájú alaplapi vezérlőáramkör-készlete, amelyet 1999-ben Whitney fantáziánéven jelentetett meg a Pentium II, III és Celeron processzorokhoz. Főbb részei a memória és grafikus vezérlő hub (GMCH = Graphics and Memory Controller Hub), az I/O vezérlő hub (I/O Controller Hub) és a BIOS-t kezelő hub (FWH = Firmware Hub).

8B/10B kódolás

Az IBM által bevezetett módszer a soros átvitelre, amely minden 8 bites adatot 10 bites továbbítandó karakterre alakít át. A 10 bites kód sok 1-0 és 0-1 átmenetet tartalmaz, ezzel könnyítve meg a bitek szinkronizálását külön átvitt órajel hiányában.

AC (Accumulator)

L.: Akkumulátor regiszter

AC97 CODEC interfész

Az Audio Codec 97 szabványnak megfelelő interfész, amely a 810-es áramkörkészlettel együtt került bevezetésre. Ezen keresztül hangfeldolgozó egységgel együtt kapcsolódhat az alaplaphoz az AMR (Audio Modem Riser) kártya, amely az alaplapon helyettesíti a hangkártyát és a modem kártyát. (Természetesen ekkor a hangkártya társprocesszora által végzett feladatokat a CPU-nak kell átvállalnia.)

ACPI (Advanced Configuration Power Interface)

PC-k és hordozható számítógépek energiafelvételére vonatkozó szabvány. Gondoskodik arról, hogy az eszköz csak szükség esetén legyen aktív, és csökkenti az órajel-frekvenciát, ha nincs szükség a maximális számítási teljesítményre.

AGP (Accelerated Graphics Port)

L.: AGP busz

AGP busz(Accelerated Graphics Port bus)= Gyorsított grafikus port

A 3D grafikus megjelenítő programok futtatását segítő Intel szabvány, amely a PCI buszhoz képest jóval gyorsabb közvetlen összeköttetést valósít meg a monitorvezérlő

kártya és a rendszermemória között. (1X=264 Mbajt/sec, 2X=528 Mbajt/sec, 4X=1056 Mbajt/sec, 8X=2112 Mbajt/sec.

AGP sín

L.: AGP busz

AGP(Accelerated Graphics Port)

L.: AGP busz

ALU (Arithmetic Logic Unit)

L.: Aritmetikai-Logikai Egység

AMD K6-2 mikroprocesszor

Az AMD első 3D grafikus képességekkel rendelkező processzora. A 3D Now! utasításkészlet bővítés 21 további SIMD utasítást is tartalmaz a 3D grafikus műveletekhez.

AMD K6-2 processzor

L.: AMD K6-2 mikroprocesszor

AMR kártya (Audio Modem Riser card)

Az AMR szabvány egy méretezhető interfészt és bővítőkártyát definiál a hardvergyártók számára, amelyek a hangfeldolgozást és a modem funkciókat támogatják. A szabvány útmutatása alapján készülő eszközöknél az analóg I/O funkciók egy külön audio és/vagy modem funkciót megvalósító kártyára kerülnek.

L.: AudioCodec '97

AND (Logikai művelet)

Két operandus között logikai „ÉS” művelet végrehajtása bitről-bitre.

ANSI (American National Standards Institute)

L.: Amerikai Nemzeti Szabványügyi Hivatal

ANSI/IEEE 754

Lebegőpontos számábrázolási szabvány, amely 1985-ben került elfogadásra. A szabvány nem csak az adatformátumokat definiálja, hanem a lebegőpontos műveletek, átalakítások végrehajtását is szabályozza.

API (Application Program Interface)

L.: Felhasználói programcsatoló

APM (Advanced Power Management)

L.: Fejlett energiagazdálkodás

AS/400

Az IBM által gyártott miniszámítógép típus.
Néhány fontosabb tulajdonsága:
- SMP architektúra;

- Párhuzamos elérési utak a processzorok, a memória és a háttértárak között;
- Moduláris felépítés, ikeregységek;
- Skálázhatóság (választhatóan 2-12 processzoros kiépítés).

ASCII kód

A számítógépekben általánosan ASCII kódot (American Standard Code for Information Interchange) alkalmazzuk az adatok ábrázolásá-ra. Az ASCII kódnál egy jelkombináció 8 bitből (1 bájtból) áll: 7 bit + 1 paritásbit.

AT (Advanced Technology) = Fejlett technológia

Eredetileg az 1984-ben bemutatott IBM PC neve, amelynek jellemző adatai: 80286-os processzor, 16 bites adat-, 24 bites címsín, 6 - 12,4 MHz órajel frekvencia. Napjainkban bármely 16 vagy 32 bites processzorral rendelkező IBM PC-kompatibilis számítógépet AT-nek szoktak nevezni.

AT Extended

L.: ATX

AT-sín

L.: Bővítőbusz

ATX(Advanced Technology Extended)

Alaplap architektúra specifikáció (Intel 1995), amely többek között meghatározza a számítógépházban a kábelek vezetését, a processzor és a memóriafoglalatok elhelyezését, a processzor hűtését.

AU(Address Unit)

L.: Címszámító és védelmi egység, tárkezelő egység

AWARD BIOS

Az egyik legelterjedtebb BIOS program, az Award Software cég terméke.
L.: BIOS

Ablaktechnika (regisztertárak) (REGISTER WINDOWING)

Regisztertár kezelésének egyik formája. Alkalmazása esetén a regisztertömb fix méretű részekre van felosztva, de ezeknek lehetnek az egyes taszkok között átlapolható részei is (azaz ez esetben egy taszk egy másik taszk regiszterkészletének egy részét is "láthatja"). A különböző taszkok közötti, regiszterekben történő paraméterátadást segíti.

Abszolút címzés

Az utasítás címrésze a műveletben részt vevő adatok, az operandusok valódi címét, azaz a memóriarekesz sorszámát tartalmazza, illetve a processzor egy regiszterére hivatkozik.

Accelerator card

L.: Gyorsítókártya

Accumulator

L.: Akkumulátor regiszter

Adapter

L.: Interfész

Adat-előreengedés (DATA FORWARDING)

A pipelining során fellépő adatütközések kezelésének hardveres módszere. Ha egy utasítás a pipelineba már korábban bekerült utasítás eredményadatára hivatkozik, akkor az adatátadást a processzoron belül megfelelő áramkörök biztosítják (adatátadás csak az első utasítás végrehajtási fázisa után lehetséges).

Adat

Tények, fogalmak, eligazítások olyan formai megjelenése (képe), amely alkalmas az emberi vagy az automatikus eszközök által történő kommunikációra, értelmezésre vagy feldolgozásra.

Adatbázis szerver (LAN/WAN)

Nagy adatbázisokban történő keresések leghatékonyabban úgy valósíthatók meg, ha a hálózaton csak a kérdés (pl. SQL parancs), illetve a válasz (kigyűjtött adatok) kerülnek átvitelre, és nem az egész adatbázis, illetve kezelő programja. Ezt a feladatot látja el az adatbázis szerver, manapság igen gyakran Web felületen keresztül.

Adatbáziskezelő szoftver

A felhasználók adattárolással és visszakereséssel kapcsolatos feladatait egységes módon megoldó programcsomag.

Adatcache

L.: Elsőszintű cache

Adatfeldolgozó rendszer

Mindazon módszerek, eljárások és eszközök szervezett együttese, mellyel a szervezet adatot fogad, rögzít, feldolgoz, tárol, továbbít és megsemmisít.

Adatfüggőség

L.: Adatütközés

Adatkiírás illetve beolvasás pufferregisztere (I/O eszközvezérlő)

Az I/O eszközök és a processzor közötti adatátvitel során az adatok átmeneti tárolását biztosító, az eszközvezérlőben található regiszter(ek). Minden egyes eszközvezérlő funkcionálisan tartalmaz ilyen átmeneti tárolót, amellyel áthidalható a processzor és a folyamatban levő I/O műveletek közötti sebességkülönbség.

Adatkonverzió (DATA CONVERSION)

Az adatábrázolás formájának megváltoztatása (például a bináris ábrázolásról a tízes vagy tizenhatos számrendszerre való áttérés).

Adatpuffer

Adatok átmeneti tárolását biztosító memóriarész, amely a legtöbb esetben a különböző sebességű részegységek közötti adatcseréhez szükséges.

Adatátbocsátó képesség

Az I/O rendszer teljesítményét jellemző érték, az időegység alatt átvitt adatok mennyisége. Mértékegysége például Mbájt/sec.

Adatátviteli utasítások = Adatmozgató utasítások

Az utasításkészletnek azok az utasításai, amellyel a tárolók között lehet mozgatni, másolni az adatokat:

- memória, regiszter (MOVE)
- memória, háttértárolók, perifériák (IN, OUT)
- regiszter, verem (POP, PUSH).

Általában az utasításkészletben megkülönböztetik:

- az egy- vagy több-bájtos átvitelt és
- a szavas (2, 4, 8 bájtos) átvitelt.

Adatütközés = Hazard, veszélyhelyzet, adatfüggőség

A pipelining működés során fellépő problémák egyike. Egy programutasítás a pipelineba már bekerült utasítás eredményadatára (memória vagy regiszter) hivatkozik.

Address Unit= AU

L.: Címszámító és védelmi egység

Advanced Power Management = APM

L.: Fejlett energiagazdálkodás

Advanced Technology

L.: AT

Akkumlátor regiszter = AC, ACC, ACCumlator (ACCUMULATOR)

A processzornak a logikai és aritmetikai műveletek végzéséhez használt munkaregiszttere. Nevét az összegek halmozott gyűjtéséről kapta.

Aktív eszköz = Master

(MASTER)

Alaplap(MAINBOARD, MOTHERBOARD)

A PC központi részegységeit (processzor, sínrendszer, memória, interfészek stb.) összefogó áramkörti kártya.

Alaplapi BIOS

L.: BIOS

Alaplapi áramkörkészlet

L.: Csipkészlet

Alaplapra integrált perifériák

Olyan I/O eszközök, amelyek vezérlőáramköreit az alaplap vezérlőáramkör-készlet (chipset) tartalmazza. Napjainkban már az alaplapra integrálnak minden olyan eszközvezérlést, amelyeket egy szokásos számítógépnek tartalmaznia kell. Ilyenek például a hajlékony- és merevlemez vezérlő illetve interfész, soros és párhuzamos

portok, PS/2 egér, USB csatoló. A BIOS Setup programjában tudjuk ezek használatát engedélyezni, ill. tiltani.

Alaplaphoz integrált

Az alaplapi vezérlőáramkör-készlet (chipset) részeként megvalósított I/O eszközvezérlés vagy csatoló.

Alfabetikus

A csak betűkből álló karaktersorozatok alfabetikusnak nevezzük.

Alfanumerikus (ALPHANUMERIC)

Olyan karakterkészlethez tartozik, amely betűket, számjegyeket és esetleg vezérlőkaraktereket, különleges karaktereket és szóközt tartalmaz.

Alfapuffer (ALPHA-BUFFER)

Memóriaterület, amelyben a számítógépes grafikában az átlátszó objektumok ábrázolásánál az átlátszó felület mögötti kép textúráját tárolják. A képernyőn való megjelenítéskor az átlátszó textúra és az Alfa-pufferben tárolt textúra színadatait átlagolják.

Algoritmus

Egy feladat megoldását eredményező véges számú lépésben véget érő, értelmű szabályokkal megfogalmazható műveletsorozat.

Alpha-buffer

L.: Alfapuffer

Alt (billentyű)

Váltóbillentyű. A bal Alt-ot lenyomva a numerikus billentyűzeten egy 0-255 közötti számot írhatunk be, majd felengedésével ez a szám ASCII karakterként értelmeződik. Így olyan jeleket is bevitethetünk, amelyek nincsenek a klaviatúrán (pl. nemzeti karakterek, amelyek 128 és 255 közöttiek: á=Alt+160, é=Alt+130 stb.). Másik szerepe a Ctrl-hoz hasonlóan billentyű-jelentésmódosítás (pl. Alt+X – Turbo Pascalnál kilépés DOS-ba stb.).

Alárendelt számítógép (SLAVE COMPUTER)

Elosztott számítógéprendszerben az egyes részfeladatokat ellátó, a felügyelő (Master) számítógépek vezérlésével működő számítógépek megnevezése.

American National Standards Institute

L.: Amerikai Nemzeti Szabványügyi Hivatal

Amerikai Nemzeti Szabványügyi Hivatal = ANSI (AMERICAN NATIONAL STANDARDS INSTITUTE)

Az ipar támogatásával 1918-ban alapított szabványügyi szervezet nevének rövidítése. Feladata, hogy kidolgozza az USA ipari szabványait, illetve ezeknek a változatait, amelyek megfelelnek az ISO ajánlásainak.

Analóg

L.: Analóg

Analóg (ANALOG)

Olyan eszköz vagy jel jelzője, amelynek (amely jellemzőinek) erőssége vagy mennyisége folytonosan változik. Ilyen például a feszültség vagy a hang.

Analóg-digitális konverzió = Analóg digitális konverzió, A/D átalakítás

Olyan eljárás, amely az analóg jelből a neki megfelelő digitális jelet általában mintavétellel és kvantálással előállítja.

Archív tár (ARCHIVAL STORAGE)

Soros elérésű háttértár, elsősorban mentési célra használják.

Arithmetic Logic Unit

L.: Aritmetikai-Logikai Egység

Aritmetikai művelet (ARITHMETIC OPERATION)

Aritmetikai szabályok szerint végrehajtott művelet. Például: összeadás, kivonás, szorzás, osztás, egész kitevőjű hatványozás.

Aritmetikai-Logikai Egység = ALU

A processzor aritmetikai, összehasonlítási és logikai műveleteket végrehajtó részegysége.

Assembly (ASSEMBLY LANGUAGE)

Olyan, rövidítéseket vagy mnemonikus kódokat használó alacsony szintű programozási nyelv, amelyekben minden utasítás egy gépi kódú utasításnak felel meg. Az Assembly-ben írt programot az Assembler fordítóprogram fordítja le a processzor által közvetlenül értelmezhető gépi utasításokra.

Asszociatív tároló

A tartalma alapján címezhető. Ha tároló bemenetére egy konkrét adatértéket írunk, akkor a tároló kimenetén megkapjuk, hogy ezt az adatértéket a tároló tartalmazza-e vagy sem. Vagyis ez olyan tárolótípus, melynél a rekeszekben tárolt információ nem a rekeszek címei alapján, hanem a tárolt információ egy része (keresési kulcs) alapján olvasható ki.

Asynchronous transmission

L.: Aszinkron adatátvitel

Aszinkron (ASYNCHRONOUS)

Olyan időbeli folyamat jelzője, amelynek lefutását nem szinkronizálja a teljes folyamatra érvényes, egységes órajel.

Aszinkron SRAM

A statikus RAM egyik típusa. Működése nincs szinkronizálva a processzoréval, ezért a processzor esetenként várakozásra kényszerül.

Aszinkron adatátvitel

Az adatátvitelnek az a módja, amikor az Adó és Vevő működését nem szinkronizálja azonos órajel. Az átvitt adatok kezdetét egy start jel, végét pedig egy stop jel jelzi.

Aszinkron sínvezérlés

A számítógép sínrendszerén történő adatátvitel vezérlésének egyik formája, amikor az Adó és Vevő egység nem használ közös órajelet. Mivel ekkor az események tetszőleges időpontokban bekövetkezhetnek, ezért a sínre csatlakozó eszközök zavartalan együttműködéséhez egy kapcsolatfelvétel és vétel visszaigazolási eljárás (handshake) szükséges. Ennek a sínvezérlésnek az előnye, hogy nagyon eltérő sebességű eszközök kiszolgálását is lehetővé teszi. Hátránya viszont, hogy a protokollba be kell építeni a relatíve bonyolult aszinkron, kettős "handshake" eljárást.

Asztali számítógép (DESKTOP PC)

A "klasszikus" PC megnevezése, amelyet különálló számítógépház és perifériák (monitor, billentyűzet stb.) jellemeznek.

Athlon processzor

Az AMD cég K7 jelű 1999-es processzorának fantázianeve.

Attrib (DOS parancs)

Fájlok és könyvtárak jellemzőit (írásvédelem stb.) nézhetjük és változtathatjuk meg.

Audio Codec '97

Ipari szabvány, amelynek célja a hangfeldolgozó hardvereszközök egységes kezelése. Fizikai megvalósítása általában az alaplaphoz integrált hangfeldolgozó egységgel történik, amely képes kezelni a hang CD, az analóg mikrofonbemeneteket, az analóg TV vevő- vagy a videojeleket feldolgozó kártyákat valamint az analóg modemeket.

BCD kód

Decimális számjegyek olyan bináris ábrázolási módszere, ahol minden decimális számjegyet helyi értékenként 4 bittel ábrázolnak. (Más szóval: a 0–9 számjegyeket 2-es számrendszerben fejezik ki, mindegyikben 4 helyi értéket véve.)

BEDO RAM (Burst Extended Data Output Random Access Memory)

Olyan memóriatípus, amely nagy sebességét a "burst" hozzáférési technika és az EDO RAM kombinációjával éri el.

BIOS (Basic Input Output System) = Központi bemeneti/kimeneti rendszer

Az operációs rendszer ROM-ban vagy EPROM-ban vagy Flash memóriában tárolt része, amely az alapvető I/O rutinokat tartalmazza. A BIOS hajtja végre a számítógép bekapcsolásakor a hardver legfontosabb ellenőrzéseit és egyes rendszerfájlok betöltését. Egyes részprogramjait (funkcióit) szoftvermegszakítás útján (INT utasítás) hívhatják fel a felhasználói programok.

BIOS setup programja

A BIOS funkcióinak beállító programja, amellyel a számítógép alpműködését, hardverbeállításait és az operációs rendszerrel való kapcsolatát szabályozhatjuk. Ez a program minden számítógépben megtalálható, gyártójától függő megjelenítési formában. Elindítása a gép bekapcsolása után a BIOS képernyőn kiírt billentyű megnyomásával

lehetséges.

BIU (Bus Interface Unit)

L.: Buszillesztő egység

BTB

L.: Branch Target Buffer

Backspace (billentyű)

(Vagy balra mutató nyíl) – szövegbeírásnál a kurzortól balra álló karakter törlésére szolgál, egyben a kurzortól jobbra eső rész balra mozog.

Batch Processing

L.: Kötegetelt feldolgozás

Baud

Az adatátvitel sebességének mértékegysége. 1 Baud sebességű az az átvitel, mely 1 másodperc alatt 1 jelet visz át. Ha az átvitt jelek bináris értékek, akkor a Baud megegyezik a bit/szekundummal.

Bekapcsolási önteszt program= POST, kezdeti betöltő(POWER-ON SELF TEST)

A számítógép csak olvasható memóriájában (ROM) található BIOS-rutinok egy része, amelyek a rendszer alkotóelemeinek (például a RAM, a lemezmeghajtók és a billentyűzet) megfelelő csatlakoztatását és működését ellenőrzik a számítógép bekapcsolásakor. A POST rutinok a hibákat hanggal vagy képernyőre küldött üzenettel üzenettel jelzik a felhasználónak.

Belső gyorsítótár = L1 cache

L.: Elsőszintű cache

Belső sínrendszer

A processzoron belüli sínrendszer, amely a processzor különböző részeit kapcsolja össze, ezek kommunikációját biztosítja. Sebessége (órajele) megegyezik a processzoréval.

Belső óra (alaplap) (INTERNAL CLOCK, REAL TIME CLOCK)

Az alaplapi CMOS RAM-nak a pontos időt és dátumot kezelő áramköre, amely a gép kikapcsolása után is működik. Tartalmát az operációs rendszer és a felhasználói programok is felhasználják. Nem azonos az órajel-generátorral.

Bemeneti/kimeneti port (INPUT-OUTPUT PORT)

Mikroprocesszorok azon funkcionális egysége, amelyen keresztül a CPU és I/O egységek közötti adatforgalom lebonyolódik.

Benchmark

1. Számítógéprendszer teljesítményét kiértékelő program.
2. Speciális teljesítménymérő programokkal előállított mutató, amely a számítógéprendszerek teljesítményéről összetett képet ad. Mindig átlagteljesítményt

fejez ki.

Betöltő (loader) program

Az operációs rendszer magjához tartozik, feladata a végrehajtható program elhelyezése a memóriában, a bázis cím kitöltése a megfelelő értékkel, a folyamatleíró blokk elkészítése. A betöltéssel válik egy program folyamattá.

Beviteli/kiviteli eszköz= I/O eszköz (I/O DEVICE)

Hardverelem, amely adatokat küld a számítógép központi egységének, vagy adatokat fogad a központi egységtől (például a lemezmeghajtó). Néhány eszköz - például a billentyűzet vagy az egér - inkább bevitelre használatos, más eszközök - mint a nyomtatók - csak kivitelre használhatók, ezért ezeket beviteli illetve kiviteli eszközöknek nevezik. A személyi számítógépek esetében I/O eszközök közé soroljuk az alaplap vezérlőáramkörök (chipset) egyes funkcionális feladatait megvalósító áramköröket is. Ilyenek: óra áramkör, időzítő/számláló áramkör stb.

Billentyűkód (KEY CODE, SCAN CODE)

A számítógépes billentyűzet egy adott gombjához rendelt kódszám, amely meghatározza, hogy melyik billentyűt ütötte le vagy engedte fel a felhasználó. A billentyűkód magának a billentyűnek a speciális azonosítója, tekintet nélkül a billentyűn található betűre, számra, szimbólumra vagy a gomb által generált karakterre.

Billentyűzet (KEYBOARD, KEYPAD)

Szöveges adatoknak és parancsoknak a számítógépes rendszerbe történő bevitelére szolgáló eszköz.

Binary

L.: Bináris

Bináris logika

Az olyan logikai rendszer, amely kétállapotú elemekkel valósítja meg a kívánt logikai kapcsolatokat.

Bináris, Kettes számrendszerbeli (BINARY)

1. Két különböző értékkel vagy állapottal jellemezhető (pl. 0 vagy 1, magas vagy alacsony jelszint stb.) 2. Olyan rögzített alapszámú számrendszerből való, amelynek alapszáma kettő.

Block transfer

L.: Blokkátvitel (DMA)

Blokktechnika (regisztertárak) (REGISTER BLOCKING)

A regisztertárak kezelésének egyik típusa. Alkalmazása esetén a regisztertömb változó méretű, átlapolható részekre van felosztva. Ez a különböző taszkok közötti, regiszterekben történő paraméterátadást segíti.

Blokkátvitel (DMA)

A processzor és a DMA között a buszhasználat megosztásának egyik módja. Blokkátvitel esetén a DMA-vezérlő az adatátvitel teljes idejére lefoglalja a buszt.

Boole-algebra (BOOLEAN ALGEBRA)

Kétértékű logikai váltakozókra definiált algebra, melyet George Boole angol matematikus dolgozott ki 1850 körül. Logikai algebrának is nevezik. Legfontosabb alkalmazása a G.E. Shannon által kidolgozott kapcsolás-algebra, mely a digitális áramkörök működésének alapját képezi.

Boolean algebra

L.: Boole-algebra

Boot

L.: Operációs rendszer betöltése

Bootolás

L.: Operációs rendszer betöltése

Branch Target Buffer = BTB

A processzorba épített cachetároló, amely a spekulatív elágazás-feldolgozáshoz szükséges adatokat tartalmazza. A feltételes ugrások dinamikus előrejelzéséhez tárolja a már végrehajtott ugróutasítások statisztikai adatait.

Bridge áramkörök

Két eltérő adatátviteli protokollt alkalmazó rendszerelem közötti adatkapcsolatot biztosító áramkörök.

Bridge

L.: Bridge áramkörök

Buffered write through

L.: Pufferelt közvetlen átírás

Bulk storage

L.: Tömegtároló-egység

Burst Modus

L.: Burst

Burst átvitel

L.: Burst

Burst üzemmód

Egyes újabb RAM-típusoknál (SDRAM, BEDO RAM, DDR SDRAM) alkalmazott, a korábbi RAM-oknál gyorsabb hozzáférést biztosító technika. Ezt a gyorsabb üzemmódot a processzor és a memória működésének szinkronizálásával, és az adatok blokkokba szervezésével érik el.

Burst

Memória-hozzáférési módszer, amelynél az adatokat megfelelő áramkörökkel blokkokba

szervezik. A burst hozzáférésnél egymás utáni címeken található adatokat lehet kiolvasni a memóriából, amihez a memóriavezérlés automatikusan generálja az egymás után következő címeket.

Bus arbitration

L.: Buszarbitráció

Bus cycle

L.: Buszciklus

Bus protocol

L.: Buszprotokoll

Busz = Busz(rendszer), sínrendszer, sín(BUS)

A számítógép részegységei közötti kommunikációs kapcsolatokat (adatok, címek, valamint a gép vezérléséhez szükséges információk átvitelét) lehetővé tevő vezetékrendszer és ennek vezérlő áramkörei.

Buszarbiter

Sínfoglalási kérelmeket fogadó, elbíráló és visszaigazoló hardveregység.

Ha a buszarbitráció központosított, a számítógéprendszerben csak egy arbiter van. Ez lehet egy önálló hardveregység, vagy egy másik hardvereszköznek (legtöbbször a CPU-nak) valamilyen részegysége.

Ha a buszarbitráció szétosztott, a számítógéprendszerben több arbiter hardveregység található.

Buszarbitráció = Sínarbitráció (BUS ARBITRATION)

Sínhasználat jogának eldöntésére szolgáló folyamat. Akkor van jelentősége, ha egyidejűleg több aktív (master) eszköz is igényelheti a sín használatát. A buszarbitráció lehet központosított vagy szétosztott.

Buszciklus = Sínciklus (BUS CYCLE)

Egy adategység sínen történő átviteléhez szükséges idő (ennek végén jelennek meg a szükséges jelszintek az áramkörökben).

Buszfoglalási kérelem = REQ (REQUEST)

A sín lefoglalását kérelmező eszköz által beállított vezérlőjel.

Buszfoglalási kérelmet engedélyező vezérlőjel = GNT (GRANT)

A sínvezérlés sínhasználatot engedélyező vezérlőjel.

Buszillesztő egység = BIU, buszinterfész egység (BUS INTERFACE UNIT)

A processzor egyik legfontosabb architektúrális építőeleme, amely biztosítja a processzor kapcsolódását a külső sínrendszerhez.

Buszinterfész egység

L.: Buszillesztő egység

Buszprotokoll (BUS PROTOCOL)

A sínrendszeren megvalósuló kommunikációs kapcsolatok szabályainak szabványosított összessége. (Működési szabályok, mechanikus és elektromos jellemzők stb.) A buszprotokollban meghatározott szabályrendszer fizikailag a buszvezérlő hardveregységben testesül meg.

Buszrendszer

L.: Busz

Buszspecifikáció

A buszrendszer architektúráját és protokollját leíró dokumentáció.

Busztranzakció

A buszigények sorozata, amely az adatátvitel igénylésétől annak befejezéséig tart (több műveletből, buszciklusból állhat).

Egy busztranzakció lépései a következők lehetnek:

- sínhasználat igénylése (REQUEST)
- sínhasználat jogának odaítélése (ARBITRATION)
- címátvitel (ADDRESSING) és adatátvitel (DATA TRANSFER)
- hibafelismerés (ERROR DETECTION)
- felszabadítás (RELEASE).

Bővítőbusz = AT-sín, bővítősín

Az IBM AT és a vele kompatibilis számítógépek által használt sínrendszer.

Bővítőkártya (DAUGHTER BOARD, EXPANSION BOARD, EXTENDER BOARD)

A számítógép alaplapijának bővítőkártya helyeibe (az ún. slot-okba) illeszthető áramkörti kártya, amely legtöbbször az I/O eszköz(ök) vezérlését és/vagy csatlakoztatását biztosítja. A napjainkban gyártott alaplapokhoz alapvetően három típusú bővítőkártya használható: ISA, PCI, AGP.

Bővítősín

L.: Bővítőbusz

C/BE vezetékek

A PCI busz bájt kiválasztó vezérlővonalai.

CAS (Column Address Strobe) = Oszlopcím kiválasztó impulzus

A rendszermemória multiplexált sor-oszlop címzéséhez szükséges vezérlőjel, amely jelzi a DRAM csipnek, hogy a megfelelő oszlopcím a címpufferbe elhelyezésre került.

CAW (Channel Address Word)

Az I/O műveletekhez szükséges címadatokat tartalmazza, a CAW a CCW-vel együtt az I/O processzorok "csatornaprogramja".

CCW (Channel Command Word)

Az I/O művelethez szükséges utasítások adatait tartalmazza, a CAW a CCW-vel együtt az I/O processzorok "csatornaprogramja".

CCW (Channel Command Word)

Az I/O művelethez szükséges utasítások adatait tartalmazza, a CAW a CCW-vel együtt az I/O processzorok "csatorna-programja".

CD-ROM (Compact Disk-Read Only Memory) = Csak olvasható optikai lemez

Csak olvasható 650 MB-ot kapacitású lemez, amelyről az adatok kiolvasása lézeroptikai eszközzel történik. A Philips és Sony cégek fejlesztették ki a multimédia számára integrált lejátszóközegnek.

CD-ROM-meghajtó (CD-ROM DRIVE)

Optikai lemezek olvasását biztosító input hardverrészegység.

CISC (Complex Instruction Set Computer)

L.: Komplex utasításkészletű számítógép

CMOS RAM (COMPLEMENTARY METAL-OXIDE-SEMICONDUCTOR RANDOM ACCESS MEMORY)

Az alaplap részegysége, egy 64 vagy 128 bájtos RAM memória, amely tárolja a számítógép hardverfelépítésének legfontosabb adatait (például mekkora RAM memória van a számítógépben stb.). Emellett tartalmazza a számítógép ún. belső óráját, amely a gép kikapcsolása alatt is működik, és szolgáltatja a pontos időt és dátumot.

CMOS setup

L.: CMOS-beállítás

CMOS óra-áramkör

L.: Belső óra (alaplap)

CMOS-beállítás = CMOS setup (COMPLEMENTARY METAL-OXIDE-SEMICONDUCTOR SETUP)

A BIOS hardverbeállításainak módosítása. Ez a ROM-BIOS-ban található Setup programmal hajtható végre a számítógép bekapcsolásakor (általában a Delete billentyű lenyomásával juthatunk be a Setup menüjébe). Itt állíthatjuk be a háttértár adatait és az alaplap jellemzőit, amely adatokat a CMOS RAM tárol.

CNR (Communication and Networking Riser)

Kommunikációs és hálózati kártya, amely hangvezérlő és hálózati funkciókat (a modemet is beleértve) megvalósító áramköröket tartalmaz.

COM port

L.: Soros port

CPU (Central Processing Unit)

L.: Központi egység

CTRL+ALT+DEL

L.: Melegindítás

CU (Control Unit)

L.: Vezérlőegység (processzor)

Cache hit

L.: Cachetalálat

Cache miss

L.: Cachetévesztés

Cache

L.: Gyorsítótár

Cachesor

A cachebe lemásolt főtárblokk és a hozzá tartozó vezérlő információk helye a cachetárolóban. Minden sor két részből épül fel:

- a todalék (tag) egyrészt a főtárból bemásolt blokkra vonatkozó címinformációkat tartalmazza, másrészt itt kerülnek bitenként kódolva letárolásra a cacheblokk adataira vonatkozó érvényességi információk;
- az adatrész tartalmazza a bemásolt főtárblokk változatlan vagy a processzor által már módosított adatait.

Cachetalálat (CACHE HIT)

Az az eset, amikor a processzor olyan adatot igényel, amely a cacheben megtalálható.

Cachetévesztés (CACHE MISS)

Az az eset, amikor a processzor olyan adatot igényel, amely a cacheben nem található meg.

Cachevezérlő

Egy "intelligens" áramkörökből felépített hardveregység, amely vezérli a cacheben történő adatvisszakeresést, intézi a főtárblokkok cachebe történő bemásolását, illetve a megváltozott blokkok visszaírását a főtárba.
(Intelligencia alatt az értendő, hogy a vezérlő képes a különböző bemásolási és adatvisszaírási stratégiák, algoritmusok kezelésére.)

Callkapu

Az Intel és vele kompatibilis processzorok tárolóvédelmi rendszerének négy kaputípusa közül az egyik. A taszkok közötti paraméterátadásra szolgál.

Caps Lock (billentyű)

Kisbetű/nagybetű váltó. Megnyomva kisbetűk helyett nagybetűket használhatunk (ekkor a Shift+betű-re kisbetűket kapunk). Helyzetét egy LED jelzi (ha ég – NAGYBETŰS üzemmód).

Cd (DOS parancs)

Aktuális könyvtár megváltoztatása.

Celeron mikroprocesszor

Az Intel 1998 áprilisában kihozott, P6 architektúrájú processzora, amely a Pentium II,III és 4 leegyszerűsített, olcsó PC-kbe szánt változata. A kedvezőbb ár elérése érdekében a Celeronok kezdetben nem tartalmaztak L2 gyorsítótárat. Későbbi változataik már 128 kb-át L2 cachesel kerültek gyártásra.

Centralizált számítógéprendszer

A számítógéprendszerek feladatvégrehajtó egységeinek térbeli elhelyezkedése szerint megkülönböztetett üzemmód.

Egy számítógéprendszer centralizált, ha csak egy központi vezérlést és feladatvégrehajtó egységet tartalmaz (ez nem zárja ki, hogy a rendszer szolgáltatásait távadatfeldolgozó hálózattal a központi számítógéprendszerből igen messze is igénybe vegyék a felhasználók).

Centronics párhuzamos interfész (CENTRONICS PARALELL INTERFACE)

A számítógép központi egysége és a perifériák közötti 8 bites output irányú párhuzamos adatátvitel szabványa, amelyet eredetileg a nyomtatógépgyártó Centronics cég fejlesztett ki.

Chipset

L.: Csipkészlet

Ciklus (CYCLE)

Utasítások olyan sorozata, amely ismételten végrehajtandó, amíg egy meghatározott feltétel fennáll.

Cikluslopás (CYCLE STEALING)

A processzor és a DMA között a buszhasználat megosztásának egyik módja, átlapoló buszhasználatot jelent a processzor és a DMA-vezérlő között.

A busz felszabadítása minden szó (bájtt) átvitele után megtörténik, ütközés esetén a DMA-nak prioritása van.

Clock multiplier

L.: Órajel-sokszorozó

Clock pulse

L.: Órajel

Cls (DOS parancs)

Képernyő törlése (Clear Screen)

Column Address Strobe

L.: CAS

Command interpreter

L.: Parancsértelmező

Communication and Networking Riser

L.: CNR

Communication

L.: Kommunikáció

Computer

Számítógép

Control Unit

L.: Vezérlőegység (processzor)

Control flow

L.: Vezérlésáramlásos vezérlés

Control signal

L.: Vezérlőjel

Copy (DOS parancs)

Fájl másolása (még egy példány készítése) más könyvtárba, esetleg más néven.
Ritkábban: állományok összefűzése, vagy új szövegfájl létrehozása.

Csatlakoztasd és használd

L.: Plug and Play

Csatoló a processzorhoz (HOST ADAPTER)

A processzor-memóriabuszt és az I/O buszrendszert összekötő interfész.

Csipkészlet = Alaplapi áramkörkészlet, lapkakészlet (CHIPSET)

Integrált áramkörökből álló részegysége az alaplapnak, amely ellátja a memóriák és az I/O rendszer egyes elemeinek vezérlését.

Csoport asszociatív cache

L.: N-utas csoport asszociatív cache

Csökkentett utasításkészletű számítógép = RISC (REDUCED INSTRUCTION SET COMPUTER)

Olyan számítógép, amely processzorának utasításkészlete csökkentett, egyszerűsített, a memória-hozzáférés csak két utasítással: memóriából való adatbetöltéssel (LOAD) és memóriába való adatírással (STORE) történhet, a műveleti vezérlés huzalozott vagy horizontális (egyszerűsített) mikroprogramvezérelt.

Ctrl (billentyű)

(Control) – Váltóbillentyű, önmagában általában nincs jelentése, nyomva tartása alatt egy másik billentyűt is le szoktak ütni. Pl. Ctrl+S – képernyő görgetés (scrollozás) pillanatnyi megállítása, Ctrl+C – program befejezése (lásd még Pause/Break), vagy Ctrl+Alt+Delete – melegindítás.

Cycle stealing

L.: Cikluslopás

Céleszköz (TARGET)

A SCSI csatolók sínrendszerén történő adatátvitel esetében a célegységet jelenti. Ekkor az adatforgalom minden pillanatban két egység között zajlik, amelyek kezdeményezőként (Initiator) és célként (Target) dolgozhatnak. (Egy készülék szerepe nem rögzített előre, az az átvitel során alakul ki. Általában minden eszköz képes az initiator és a target szerepet ellátni.)

Cím számító és védelmi egység = AU, MMU (ADDRESS UNIT, MEMORY MANAGEMENT UNIT)

A processzor egyik legfontosabb architektúrális építőeleme. Az Intel és vele kompatibilis processzorokban az MMU megnevezése. Feladata a programutasításokban található virtuális címek leképezése a főtár fizikai címekre, és a tárolóvédelmi hibák felismerése.

Cím számító és védelmi egység

A processzor egyik legfontosabb architektúrális építőeleme. Az Intel és vele kompatibilis processzorokban az MMU megnevezése. Feladata a programutasításokban található címek leképezése a főtár fizikai címekre, és a tárolóvédelmi hibák felismerése.

Címzési mód (ADDRESSING MODE)

A tárban vagy egyéb tárolóhelyen lévő, adatot vagy utasítást tartalmazó rekeszek helyének meghatározási módszere. Tehát az a címmeghatározási módszer, ahogyan egy alacsony szintű nyelven egy utasításban címet meg lehet adni. Olyan számítási szabályként is értelmezhető, amelynek alkalmazásával az utasításban szereplő adatokból (pl. bázisregiszter sorszáma, eltolás értéke) megkapjuk a tényleges tárcímet.

DDR SDRAM (Double Data Rate Synchronous Dynamic Random Access Memory) = Kétszeres

Az SDRAM újabb változata, kétszeres adat átbocsátóképességű szinkronizált dinamikus RAM.

DIMM (Dual Inline Memory Module)

A jelenleg alkalmazott memóriamodulok 64 bites szervezésű típusa, 168 érintkezője van (a kártya foglalatában két érintkező sor van).

DMA (Direct Memory Access)

L.: Közvetlen memória-hozzáférés

DMA I/O művelet

DMA által vezérelt I/O művelet. A DMA busmasterként a processzortól "függetlenül" vezérli az adatátvitelt a buszon, ezalatt a processzor más feladatokat hajthat végre.

DMA-adatátviteli eljárás típus

A CPU és a DMA-vezérlő közötti buszhasználat megosztásának módja. Lehet: blokkátvitel, cikluslopás, transzport mode, CPU-leállítási eljárás, memória-időszelet eljárás.

DMA-címregiszter

A DMA-vezérlő egyik regisztere, amely az adatok DMA-átviteli lépésenként aktualizált főtárbeli címét tartalmazza.

A DMA-vezérlő ebben helyezi el az átvitel előtt a processzortól kapott memóriacímeket.

DMA-maszkregiszter

A DMA-vezérlő egyik regisztere, amely az egyes DMA-átvitelt kérő vezérlővonalakat (DREQ) letiltó biteket (maszkolás) tartalmazza.

DMA-számlálóregiszter

A DMA-vezérlő egyik regisztere, amely a DMA I/O művelet során átvitt szavak számát tartalmazza. Minden átvitel után a DMA-vezérlő aktualizálja (csökkenti) tartalmát. Ha a számláló tartalma 0, a DMA-átvitel befejeződött.

DMA-vezérlő (DMA-CONTROLLER)

A közvetlen memória-hozzáféréssel történő I/O adatátvitelt vezérlő hardvereszköz. Az adatátvitelt a memória és az I/O eszköz között önállóan irányítja, miután megkapta a processzortól egy I/O művelethez a szükséges adatokat. Ezáltal a processzor felszabadul más feladatok végrehajtására.

DMA-vezérlőfunkció

Az adatátvitel végrehajtásához ellátandó vezérlési feladatok összessége. A DMA vezérlőfunkciói a következők:

- Címvonalevezérlés: a DMA-t tartalmazó rendszerekben a memóriacímbuszt vagy a CPU, vagy a DMA hajtja meg attól függően, hogy az adott ciklusban a memóriát melyik eszköz használja. A DMA-ciklusban a DMA-vezérlőnek a kívánt DMA-művelet elvégzéséhez szükséges címet kell a címbuszra adni.
- Adatátviteli vezérlés: a DMA-vezérlőnek a memória és az I/O készülék közötti közvetlen adatátvitelhez - megfelelő időzítéssel - vezérlőjeleket kell szolgáltatni. E vezérlőjelek csak a DMA-ciklusban kerülnek a vezérlőbuszra.
- Cím tárolás: a DMA-vezérlő címregisztere tartalmazza a következő írásra vagy olvasásra kerülő adat címét. Ezt minden átvitel után inkrementálni vagy dekrementálni kell.
- Adatszámolás: a DMA-adatátvitel indításakor a CPU betölti a DMA-vezérlő számlálóregiszterébe az átvitelre kerülő rekeszek számát. A DMA-adatátvitel alatt a DMA-vezérlő számlálja az átvitt rekeszeket, és a megadott számú adat átvitele után befejezi az adatátvitelt.
- Üzem módvezérlés: a DMA hardver-üzem módvezérlő regiszterét a CPU tölti fel az adatátvitel előtt. Meghatározza az adatátvitel irányát (memóriába írás vagy olvasás).

DMA-állapotregiszter

A DMA-vezérlő egyik regisztere, amely a vezérlő állapotával kapcsolatos információk tárolására szolgál.

DRAM (Dinamic Random Access Memory)

L.: Dinamikus tár

DRAM modul

DRAM: dinamikus memóriatípus, amely az adatokat csak egy meghatározott (rövid) ideig képes tárolni, ezért a memóriát üzemszerűen, periodikusan frissíteni kell (újra kell írni).
Modul: Meghatározott, 2-hatvány darabszámú memóriacsipből összeállított memóriaegység, amelyet különálló sokérintkezős nyáklapra szerelnek, és az alaplap memóriaillesztő részébe helyeznek.

DRDRAM

L.: Direct Rambus DRAM

Daisy chain (sínrendszerek)

L.: Soros buszfoglalási eljárás

Data buffer

L.: Adatpuffer

Data conversion

L.: Adatkonverzió

Data forwarding

L.: Adat-előreengedés

Date (DOS parancs)

Karbantartó parancs: a gép belső órájának (év-hónap-nap-napnév) kiírása és megváltoztatása.

Daughterboard

L.: Bővítőkártya

Debug

L.: Nyomkövetés

Defrag (DOS parancs)

Karbantartó parancs: a háttértár állományait fizikailag folytonossá rendezi.

Del (DOS parancs)

Fájlok törlésére szolgál. del. = az alkönyvtár összes fájljának törlése.

Delete (billentyű)

Szövegszerkesztő programokban a kurzor "előtt" álló karakter törlődik és a sor többi tagja balra csúszik (azaz egy Delete gomb helyettesíthető egy kurzor jobbra és egy Backspace-szel). Egyes számítógéptípusokon bekapcsolás után a Delete billentyűvel léphetünk be az ún. Setup-ba.

Deltree (DOS parancs)

K - teljes könyvtárakat töröl, tartalommal együtt. Veszélyes parancs!

Demand fetching

Az egyik leggyakoribb cachebe való blokkbemásolási eljárás. Alkalmazása esetén csak a processzor adatigénye esetén keresik ki a főtárból a megfelelő blokkot és töltik be a cachebe. Ezzel párhuzamosan a processzor is azonnal megkapja az adatot (load through).

Desktop PC

L.: Asztali számítógép

Deszkriptor (DESCRIPTOR)

A rendszer védelmi táblázataiban található adat, amely meghatározza a rendszerobjektumokhoz tartozó jogosultságokat.

Dialógus üzemmód

Az interaktív feldolgozás egyik altípusa, amikor a felhasználó kérdez-felelek módon, lépésenként folyamatosan kommunikál a programfolyamattal. Ennek tipikus esete a terminál üzemmód, amikor a felhasználó képernyőn közli a rendszerrel a parancsokat és adatokat, és a rendszer a válaszokat szintén képernyőn jeleníti meg.

Digital representation

Digitális ábrázolás

Digital

L.: Digitális

Digitális (DIGITAL)

Számjegyekkel ábrázolt; pl. olyan adatok, eszközök, eljárások jelzője, amelyek változó mennyiségeket diszkrét módon, számjegyekkel ábrázolnak, illetve dolgoznak fel.

Digitális-analóg konverter

L.: Digitális-analóg átalakítás

Digitális-analóg konverzió

L.: Digitális-analóg átalakítás

Digitális-analóg átalakítás = Digitális-analóg konverzió, D/A átalakítás (DIGITA

Eljárás, amely a digitális jelből előállítja az analóg jelet.

Dinamikus buszhasználat

Buszarbitráció egyik módja, amelynél a mesterek csak akkor kapják meg a sín használati jogát, ha azt igénylik.

Dinamikus buszhasználat csak akkor lehetséges, ha

- minden master jelzi a buszfoglalási igényét az arbiternek;
- egy master csak akkor használhatja a buszt, ha igényének elfogadását visszaigazolják;
- a masternek jelzést kell küldenie, ha a tranzakció befejeződött.

Dinamikus elágazás előrejelzés (pipeline)

A program futása közben a processzor egy táblázatban vezeti az ugróutasítások címeit és az ezekre történő ugrás gyakoriságát, és ezt felhasználva próbálja megjósolni az elágazások lehetséges kimenetét.

Dinamikus tár = Regeneratív tár

(REGENERATIVE MEMORY, REGENERATIVE STORE, DYNAMIC STORAGE, DYNAMIC MEMORY, CHESHIRE CAT STORE)

Tár, amely az információt csak egy meghatározott (rövid) ideig képes tárolni. Ezért a tárolt információt üzemszerűen, periodikusan frissíteni kell (újra kell írni), különben az elvész.

Dir (DOS parancs)

Egy könyvtár tartalmát jeleníti meg, fájlokat/könyvtárakat kereshetünk (dir/s)

Direct Rambus DRAM = DRDRAM

Csatornaorientált DRAM memória-architektúra, amely három fő részből áll: Rambus vezérlő, Rambus csatornák és maga a szűkebb értelemben vett adattároló DRDRAM.

Direct mapping cache

L.: Közvetlen leképezésű cache

DirectX

Windows 95-ös szoftver, amely az alkalmazások számára közvetlen elérést biztosít a számítógép hang- és grafikushardver eszközeihez.

Disc access time

L.: Hozzáférési idő (mágneselem)

Disc cache

L.: Hardveres lemezgyorsítás

Disc interface

L.: Lemezinterfész

Display screen

L.: Képernyő

Doskey (DOS parancs)

A prompt használatát megkönnyítő segédprogram.

Double word

L.: Dupla szó

Driver

L.: Eszközmeghajtó

Drystone

Rendszerprogramozási környezetet reprezentáló, teljesítménymérő program, illetve ezzel előállított mutató.

Dual Inline Memory Module

L.: DIMM

Dupla szó (DOUBLE WORD)

A processzortól függő (például 2 bájtos vagy 4 bájtos) szóhosszhoz képest kétszeres hosszúságú tárolóegység megnevezése, amely megfelelő utasításokban önállóan címezhető.

Duron

Az AMD cég által - a Celeron konkurenciájaként - 2000-ben piacra dobott egyszerűsített, olcsó processzor. A Duron-nak 600 MHz a legkisebb órajele, 128 Kb-ot első- és 64 Kb-ot másodszintű cachet tartalmaz.

Durva szemcsés feladatkiosztás

Multiprocesszoros architektúrában a processzorok közötti feladatkiosztás egyik típusa. Az egyes processzorok nagyobb, önálló feladatokat oldanak meg (kis mértékű kommunikációs igény).

Dynamic Random Access Memory

L.: Dinamikus tár

ECC (Error Correction Circuit)

L.: Hibajavító áramkör (memória)

EDO RAM (Extended Data Output Random Access Memory)

Dinamikus RAM típus.

Jellemzője, hogy az adat a memória kimenetén hosszabb ideig áll rendelkezésre, így a processzor gyorsabban képes olvasni a memóriát.

EDO

L.: EDO RAM

EFLAGS regiszter = Állapotjelző regiszter

Az Intel és vele kompatibilis 32 bites processzorok állapotjelző regisztere.

Fontosabb biteinek jelentése:

C (carry) bit akkor kap 1-es értéket, ha a legfelső helyiértéken átvitel keletkezik;

P (parity) bit az eredményben lévő 1-esek párosságát jelzi;

Z (zero) bit az eredmény nulla értékét jelzi;

S (sign) bit az eredmény negatív voltát jelzi;

O (overflow) bit a túlszordulást jelzi.

EIDE (Enhanced Integrated Device Equipment) = Továbbfejlesztett integrált eszköz-elektronika

Az IDE továbbfejlesztése, merevlemez-meghajtó szabvány, amely maximum 8,4 Gb-ot képes tárolni, és a LBA címezéssel kezeli a lemezeket.

EIDE csatoló

Adapter, amely összekapcsolja az I/O buszt az EIDE merevlemez-vezérléssel.

EISA (Extended Industry Standard Architecture)

L.: ISA busz

EISA busz (Extended Industry Standard Architecture) = Kiterjesztett ipari architektúra

Több számítógépgyártó által specifikált 32 bites színszabvány. Meghatározza a bővítmények (monitorvezérlő, hangkártya, meghajtóvezérlők stb.) PC-alaplapokhoz való csatlakoztatását. Az EISA felülről kompatibilis az ISA-val, ma már nem használják.

EPROM (Erasable Programmable Read-Only Memory) Erasable Programmable Read-Only M

Olyan nem felejtő memórialapka, amelynek gyártása során feltöltött tartalma speciális eljárással törölhető és programozható (ultraibolya fénnel való megvilágítás). Bár költségesebb a PROM-nál, mégis gazdaságosabb, ha sokszor kell változtatni tartalmát.

Echo (DOS parancs)

Szövegek kiíratására szolgáló parancs (batch fájlokban).

Edit (DOS parancs)

A DOS szövegszerkesztő programja.

Editor

Szövegek szerkesztését végző program.

Egyszerűsített handshaking

L.: Kézfogás

Egér(MOUSE)

Relatív helyzetadatok (koordináták) bevitelére általánosan elterjedt pozícionáló input eszköz.

Az egér lapos felszínén (például asztallapon) történő mozgatásával a felhasználó rendszerint egy kurzort vezérel a grafikus képernyőn, amelyen az objektumokat vagy parancsokat az egér gombjainak megnyomásával (kattintás) választja ki.

Egérkattintás (CLICK)

Az egér gombjának lenyomása, majd elengedése, az egér mozgatása nélkül. Az egérkattintás segítségével legtöbbször valamilyen funkciót választhatunk ki a képernyőn, például programot, esetleg programrészt aktivizálhatunk.

Elektronikus levelezés (LAN/WAN)

Elektronikus üzenet továbbítása, esetleg csatolt állományokkal együtt [SMTP, POP3, TCP/IP].

Elosztott számítógéprendszer

A feladatvégrehajtó egységek térbeli elhelyezkedése szerint megkülönböztetett számítógéprendszer üzem mód.

Több, térben esetleg távoli, önálló vezérléssel rendelkező és feladatvégrehajtásra képes számítógépet tartalmaz, amelyeket egy hálózat köt össze.

Elsőszintű cache = L1, belső gyorsítótár (LEVEL 1 CACHE)

A processzorba épített (on-chip) gyorsító tároló.

Elődekódolás

A szuperskalár processzoroknál a párhuzamos dekódolás miatt megnövekedett feladatok időigényének csökkentését célzó eljárás. 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.

Emuláció

Az az eljárás, amellyel egy program vagy hardverrészegység egy másik működését utánozza.

Energiatakarékos üzemmód

A huzamosabb ideig nem használt számítógép nagy energiafogyasztó részegységeinek kikapcsolása, amelyet a tápellátás szoftveres vezérlése tesz lehetővé.

Enter (billentyű)

A beírt utasításokat, parancsokat az Enter billentyűvel küldjük el a számítógépnek végrehajtásra. Szövegszerkesztő programoknál a sor végének jelzésére és új sor megkezdésére szolgál, a Winword (Write, WordPad) esetében a bekezdés végét jelzi.

Error Correction Circuit

L.: Hibajavító áramkör (memória)

Esc (billentyű)

Ennek a billentyűnek a leütése a legtöbb program esetében, ahogy elnevezése is mutatja (escape – menekülés) valamilyen befejezést, menüből való kilépést okoz. Másrészt az Esc egy olyan vezérlőkódot is jelent (a 27-est), amely egy kódsorozat első karaktere. Ilyen vezérlőszekvenciával (Esc-szekvencia) adhatunk parancsokat különböző perifériáknak, pl. a nyomtatónak.

Eseményorientált I/O

Az input/output igények egyik típusa. Fő jellemzője, hogy az I/O művelettel a számítógépes rendszernek külső eseményekre kell reagálnia (például egér helyzetének megváltoztatása, folyamatvezérlés). Emiatt e rendszerekben a reakcióidő, valamint az időegység alatt feldolgozható eseményszám a legfontosabb teljesítményjellemző.

Eszközkezelő BIOS

A BIOS a memóriában elhelyezkedő eszközkezelő rutinyűjtemény, amely az operációs rendszer betöltését követően a hardver vezérléséhez egy funkcióhalmazt biztosít az operációs rendszer és a felhasználói programok számára. Ezt BIOS futtatási szolgáltatásnak (Runtime Service) is nevezik.

Ezeknek az eszközkezelő 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.

Eszközmeghajtó= Eszközkezelő(DEVICE DRIVER)

A számítógép-rendszernek egy konkrét típusú hardverhez történő illesztését biztosító program, amelyet a hardvergyártó az eszközzel együtt szállít. Esetenként az eszközmeghajtó program a kártyán helyezkedik el egy ROM-ban.

Exeption

L.: Kivétel

Expansion board

L.: Bővítőkártya

Extended Data Output RAM

L.: EDO RAM

Extender board

L.: Bővítőkártya

FLAGS regiszter

Az i8086-os processzor állapotjelző regisztere.

FLAGS regiszter

L.: Állapotjelző regiszter

FPM RAM (Fast Page Mode Random Access Memory)

L.: Gyors lapmódú RAM

FPU Floating Point Unit

L.: Lebegőpontos processzor

FSB (Front Side Bus)

A processzort a rendszermemóriával összekötő sínrendszer.
Az FSB sebességéből állítják elő szorzással a processzor órajelét. Az FSB sebességével fut az alaplapi csipkészlet és a RAM is.

FTP szerver (WAN)

Fájlok másolása (elsősorban) távoli szerverek és kliensek között. [FTP, TCP/IP]

FWH (Firmware Hub)

Az alaplapi vezérlőáramkör-készlet (chipset) része (Intel 810E típusjelű csipkészlet), amely a Flash-memóriában tárolt BIOS kezelését biztosítja, és egy fizikai véletlenszám-generátort tartalmaz.

Fast Page Mode RAM

L.: FPM RAM

Fault

L.: Hiba

Fc (DOS parancs)

Ezzel a paranccsal két fájlt összehasonlíthatunk, megjeleníthetjük a különbségeket

Fdisk (DOS parancs)

A merevlemez partícióinak beállítása. Vigyázat: a háttértár adatait megsemmisíthetjük!

Fejlett energiagazdálkodás = APM (ADVANCED POWER

MANAGEMENT I)

A tápellátás szoftveres ellenőrzése. Alkalmazói programinterfész, amely a szoftveres vezérlést lehetővé teszi.

Feladat

L.: Taszk

Felbontás (RESOLUTION)

A nyomtató vagy a monitor képalkotásában a részletek finomságára jellemző adat. A nyomtatók sűrűn egymás mellé helyezett apró képpontokból alkotják a képet, felbontásukat az egy hüvelykre eső pontok számában, dpi-ben (dot per inch) mérik. A monitorok felbontásán a vízszintesen és függőlegesen megjelenített képpontok (pixelek) számát értjük.

Felhasználói program interfész (APPLICATION PROGRAM INTERFACE)

L.: Felhasználói programcsatoló

Felhasználói programcsatoló = API (APPLICATION PROGRAM INTERFACE)

A grafikus programcsomag (például Direct 3D, Open GL) és a felhasználói programrendszer között helyezkedik el a felhasználói programcsatoló. Ez általában azt is biztosítja, hogy a grafikus programcsomaggal különböző programnyelveken is kapcsolatot tarthassanak a felhasználói programok (language binding).

Felhasználói programok

Az egy-egy konkrét alkalmazás céljára kifejlesztett programokat felhasználói programoknak nevezzük.

Felügyelő számítógép (MASTER COMPUTER)

Elosztott számítógéprendszerben a részfeladatokat ellátó számítógépek működését vezérlő számítógép megnevezése.

Find (DOS parancs)

A bemenetben megadott szövegben szövegrészletet keres, az eredményt a kimenetre írja.

Finom szemcsés feladatkiosztás

Multiprocesszoros architektúrákban a processzorok közötti feladatkiosztás egyik típusa. Az egyes processzorok kisebb, egyszerű feladatokat oldanak meg (erős kommunikációs igény).

Firmware

A számítógépek általános, gyakran ismétlődő vezérlési feladatait végző programokat legtöbbször kisebb, csak olvasható memóriákban (ROM – Read Only Memory) helyezik el. Ezeket a programokat, a tárolóeszközzel együtt főmvernek (firmware) nevezik.

Fixpontos szám

Fixpontos számtárolási formánál a szám kettes számrendszerbeli együtthatóit tároljuk helyi értékeiknek megfelelően egy rögzített nagy-ságú memóriaterületen. Ez leggyakrabban

- 1 bájt méretű,
- 2 bájt méretű szó (WORD),
- 4 bájt méretű duplaszó (DWORD).

Flash tároló

L. Flash-tároló

Flash-ROM

L.: Flash-tároló

Flash-memória

L.: Flash-tároló

Flash-tároló = Flash-memória (FLASH-ROM)

A "nem felejtő" félvezető memóriáknak egy típusa, amelynek alapja az EPROM technológia. A flash-memória elektronikusan törölhető és blokkonként olvasható, valamint nincs szüksége tápellátásra, tartalmát e nélkül is megőrzi. A flash memóriák sebessége több nagyságrenddel kisebb a RAM-oknál, ugyanakkor nagy megbízhatóságúak, érzéketlenek a rázkódásra és kicsi az áramfelvételük. Ezért elsősorban a hordozható számítógépekben háttértárként és a BIOS tárolására alkalmazzák ezt a memóriatípust.

Floating Point Unit

L.: Lebegőpontos processzor

Flopiegység

L.: Hajlékonylemez meghajtó

Folyamatvezérelt üzemmód (PROCESS CONTROL MODE)

Az interaktív feldolgozási üzemmód egyik altípusa. A "felhasználó" egy gépi folyamat, amit a számítógép vezérel. Ilyenek például az ipari folyamatok irányítását, forgalmi rendszerek vezérlését, stb. végző számítógépes rendszerek, amelyek különböző mérőműszerektől, érzékelőktől érkező jeleket fogadnak és dolgoznak fel.

Fordítóprogram

Az a program, mely a szimbolikus programnyelveken megfogalmazott utasításokból a processzor számára gépi kódként értelmezhető bitsorozatot állít elő.

Forgási idő (mágneselem)

Az az idő, amíg a mágneslemez forgásával a megcímezett szektor az író/olvasófej alá ér.

Format (DOS parancs)

A háttértáron kialakítja a DOS formátumnak megfelelő fájlstruktúrát.

Forráskód (SOURCE CODE)

Assembly vagy magas szintű programnyelv írt számítógépprogram, melyet felhasználása előtt még le kell fordítani a gépi (bináris) utasításokra, azaz tárgykódú programmá.

Frame buffer

L.: Képmemória

Front Side Bus

L.: FSB

Fully associative cache

L.: Teljesen asszociatív cache

Fully interlocked handshaking

L.: Kétszeres kézfogás (sínrendszerek)

Future I/O

Az Adaptec és a 3Com által kidolgozott, a Compaq, IBM, HP által támogatott sínrendszer specifikáció, amelyet 1999 decemberében fogadtak el.

Fájl I/O

Az input/output igények egyik típusa, amelynek jellemzője a soros hozzáférés nagy adathalmazokhoz. Emiatt a legfontosabb teljesítményjelzője az időegység alatt átvitt adatmennyiség.

Fájlszerver (LAN)

Célja állományok gyors elérése abból a célból, hogy azt a kliens memóriájába betölthessük (tehát nem elsősorban a másolás), program esetén futtathassuk. [IPX/SPX, TCP/IP].

Főtár (memória)

A műveletek elvégzéséhez szükséges adatokat és programokat, valamint az eredményt tárolja későbbi felhasználás céljából.

GDT (Global Descriptor Table)

L.: Globális deskriptortábla

GMCH (Graphics and Memory Controller Hub)

Az MCH memória-vezérlőáramkörök kiegészítése a chipsetbe integrált egy 3D gyorsító áramkörökkel (raszterprocesszor).
A GMCH nem alkalmaz külön grafikus memóriát, hanem a rendszermemóriát használja.

GNT (Grant)

L.: Buszfoglalási kérelmet engedélyező vezérlőjel

GUI (Graphical User Interface)

L.: Grafikus felhasználói felület

Gajski-Kuhn féle Y diagram

A hardver rétegmodelljének egyik szemléltetési formája.
Az egyes rétegeket és ezek egymáshoz való viszonyát három fő szempont szerint

mutatja:

- strukturális felépítés: amellyel azt írjuk le, hogy az egyes rétegekhez tartozó hardverelemek milyen alacsonyabb szintű építőelemekből és milyen szerkezetben épülnek fel;
- viselkedési modell: amely meghatározza az egyes rétegekhez tartozó építőelemek működésének algoritmusait;
- fizikai, geometriai jellemzők: amellyel az egyes szintekhez tartozó hardverelemek jellemző formáját, méretét, stb. írjuk le.

General register

L.: Általános regiszter

Global Descriptor Table

L.: Globális deskriptortábla

Globális deskriptortábla = GDT (GLOBAL DESCRIPTOR TABLE)

A rendszerszintű deskriptorokat tartalmazó táblázat a processzor védett üzemmódjában. Minden feladat által elérhető táblázat, amely bármilyen objektumhoz tartozó deskriptort tartalmazhat, kivéve a megszakítás (kivétel) kiszolgáló rutinok deskriptorait. A globális tábla kezdőcímét és a táblázat méretét a GDTR regiszter tárolja.

Grafikus felhasználói felület = GUI (GRAPHICAL USER INTERFACE)

A rasztergrafikus ablakozó felhasználói felület szabványa.

Grafikus processzor (RASTER IMAGE PROCESSOR)

A monitorvezérlő kártyán található 3D "gyorsító" csip, amelynek feladata a 3D vektoros adatok átalakítása raszteres (képpontokból álló) képernyőn megjeleníthető formába (vektor-raszter konverzió).

Grant

L.: Buszfoglalási kérelmet engedélyező vezérlőjel

Graphics (DOS parancs)

Rezidens program, amely a grafikus üzemmódú képernyő kinyomtatását végzi el (aktiválása: PrintScreen).

Gyors lapmódú RAM = FPM RAM (FAST PAGE MODE RANDOM ACCESS MEMORY)

Dinamikus RAM típus, amelynél ha az áramkörök egyszer megkapják a sor és oszlopcímét, és a memóriarekesz kiolvasása megtörténik, akkor címfolytonos következő olvasásnál az áramköröknek elegendő csak a következő oszlopcímét megadni.

Gyorsítókártya (ACCELERATOR CARD)

Olyan nyomtatott áramköri kártya, amely bizonyos feladatok végrehajtásában tehermentesíti a számítógép főprocesszorát, így rendszerszinten nagyobb teljesítményt eredményez.

Gyorsítótár(CACHE)

A gyorsítótárak az utasítások és/vagy adatok átmeneti tárolására szolgáló, viszonylag kisméretű, gyors memóriapufferek, amelyek a főtár blokkjainak másolatát tartalmazzák, önálló vezérléssel rendelkeznek és a felhasználó számára láthatatlanok (elérhetetlenek).

A cache fogalmának meghatározásában blokk alatt a főtár rekeszeinek olyan egymás utáni sorozatát értjük, amelynek bemásolása a cachebe egy lépésben megtörténhet.

Gépi ciklus

Az az időtartam, amely egy processzor-alapművelet végrehajtásához szükséges. Egy gépi ciklus két fázisból áll:

- az utasítás-kihozási ciklus, mely alatt (az utasításszámláló által) meghatározott tárolóhelyről a processzor kiolvassa az utasítást;
- az utasítás végrehajtásának ciklusa.

Gépi kód (MACHINE CODE)

Az utasítások ábrázolása a gép által közvetlenül értelmezhető bináris alakban.

HIT hozzáférési idő = HIT-HF

Az adatok átlagos hozzáférési ideje a cacheben találat (cache hit) esetén.

HIT-RATE

A cachetárolóhoz való hozzáféréseknek az a százaléka, amikor az adat a cacheben megtalálható.

HOST

A hálózatban működő (nagy)számítógép megnevezése.

Hajlékonylemezes meghajtó

Cserélhető hajlékonylemezek befogadására, azok írására, olvasására szolgáló periféria.

Hajlékonylemezes tár= Flopidiszk

(FLOPPY, FLOPPY-DISK, FLOPPY-DISK SYSTEM, DISKETTE)

Mágneses jelrögzítéssel működő tárolóeszköz, amely az adatokat hajlékony anyagból készült lemezen tárolja.

Hajlékonylemez meghajtó(FLOPPY DRIVER)

Elektromechanikus eszköz, amely a hajlékonylemezre mágneses jelek felírását és a felírt jelek olvasását végzi.

Handshaking

L.: Kézfogás

Hangfeldolgozó társprocesszor

Önálló hangfeldolgozásra alkalmas processzor a hangkártyán, amely tehermentesíti a processzort. Utasítások, parancsok küldésével "felprogramozható", ezután teljes önállósággal vezérli az egyes részfeladatok végrehajtását.

Hangkártya (alaplap)

Opcionális alaplapi részegység, amelynek segítségével lehetőség nyílik hanganyag, esetenként CD minőségű sztereó lejátszására. A számítógépből érkező digitális információt analóg hangjelekké alakítja át, ami felerősítés után a hangszóróval megszólaltatható.

Hardver rétegmodellje

A hardver-architektúra leírásának egyik módja, amely az architektúra 5 szintjét különíti el:

- Rendszerszint (System, Architectural Level): a számítógép hardverét félautomatikus fő-részegységeivel írja le (például processzor, interfész, tároló stb.). A részegységeket funkcionálisan és protokolljaikkal jellemzi (például processzor esetében az utasításkészlettel.)
- Algoritmikus szint (Algorithmic Level): megadja 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 bitsorozatokat értelmezik.
- Funkcionális blokkok szintje (Functional Block, Register Transfer Level): itt határozzák meg a hardvermodulok funkcionális blokkokból (például regiszterek, számlálók, összeadók stb.) való felépítését, valamint ezek műveletvégzését (például összeadás) és a köztük való adatátvitelt. Itt már figyelembe veszik a jelek időbeli ütemezését (órajel).
- Logikai szint: ezen a szinten a hardvert kapuáramkörökkel specifikálják, ennek eszköze a Boole-algebra. Strukturálisan ez kapuáramkörökből felépülő hardvert jelent.
- Áramköri szint: ezen a szinten a jeleket már folytonos mennyiségeknek tekintjük (feszültség, áramerősség stb.). A szintnek megfelelő strukturális építőelemek a tranzistorok, kondenzátorok stb.

Hardver

Lényegében ez a számítógépet alkotó, kézzel fogható eszközök összefoglaló neve, a számítógép elektronikus áramköreit, mechanikus berendezéseit, kábeleit, csatlakozásait és perifériáit nevezzük így. A hardver eredeti angol jelentése „kemény áru” (vas-áru).

Hardveres lemezgyorsítás

A főtár (RAM) és a mágneslemez között alkalmazott gyorsítótár (cache) használata. Működésének lényege:

- ha a processzor adatokat akar kiírni a lemezre, akkor a gyorsító vezérlője ezeket letárolja a cacheben 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 lemezzől, a gyorsító vezérlője megvizsgálja, hogy az adatok megtalálhatók-e a cacheben. Ha igen, akkor az adatokat közvetlenül a cacheből továbbítja, lemezművelet végrehajtása nélkül.

Hardveres lemezgyorsító tár (DISC CACHE)

A főtár és a mágneslemez közé beiktatott cachetár, amely általában a lemezvezérlő egységbe kerül beépítésre.

Hardverleíró nyelv = VHDL (VHSIC HARDWARE DESCRIPTION LANGUAGE)

Nemzetközi szabványként elfogadott (IEEE-1076, 1987 és ANSI 1989) magas szintű hardverleíró nyelv, amelyet a hardver rétegmodelljének megfelelően végrehajtott hardvertervezésben használnak.

A VHDL hardverleírónyelv három alapegysége:

- architektúra - "Architektúra" leírásokban kell megadni egy hardveregység funkcióit és szerkezetét.
- interfész - Ez tartalmazza a hardveregységek közötti kommunikáció (protokoll) jellemzőit.
- konfiguráció - Az Architektúrák és Interfészek egymáshoz rendelését határozza meg.

Hardvermegszakítás(HARDWARE INTERRUPT)

Olyan megszakítás, amelyet a processzoron kívüli külső hardvereszköz generál. A külső hardvermegszakítások olyan esetekben keletkeznek, amikor például egy portra egy feldolgozásra váró karakter érkezik, egy lemezmeghajtó kész egy blokknyi adat továbbítására, vagy a rendszeróra meghatározott számú ciklusa lejár, vagy a hardver hibásan működik (például paritáshiba).

Hardverműködés távoli felügyelete

Lehetővé teszi a rendszergazdák számára, hogy a felhasználók számítógép-rendszerébe hálózaton keresztül beavatkozzanak, és például egy hibát kijavítsanak.

Hardware interrupt

L.: Hardvermegszakítás

Hardware monitoring

L.: Működési felügyelet

Harvard architektúra

Olyan számítógéprendszer, amelynél a programutasításokat és az adatokat fizikailag különálló memóriában tárolják. Például Pentium processzor család L1 szintű különálló adat- és utasításcache-e.

Hatodik generációs processzor

P6 architektúrájú RISC processzormaggal, RISC mikROUTASÍTÁS tárolókkal működő proceszor.

Hazard (pipelining)

L.: Adatütközés

Helyettesítési stratégia (cache)

Algoritmus, amellyel eldönthető, hogy a cachebe bemásolandó új főtárblokk adatai a cache melyik sorába kerülhetnek be (azaz az új blokk adataival melyik más, régebben a cachebe másolt blokk adatait lehet felülírni). A cache vezérlésében leggyakrabban alkalmazott helyettesítési eljárás az LRU stratégia.

Helyi hálózat

L.: LAN

Helyi sín (LOCAL BUS)

A külső sínrendszer egyik fajtája, amely a processzorhoz közvetlenül kapcsolódó gyors rendszerelemeket (memória, grafikus kártya stb.) köti össze. A helyi sínen keresztül az adatátvitel a processzor órajelével szinkronban történik, és a buszon történő adatátvitel szélessége is megfelel a processzor működésének.

Hiba = Meghibásodás (FALIURE, FAULT)

1. Egy rendszer téves működése, valamilyen hiányosság szoftver- vagy hardverműködési hiba következtében.
2. Egy program lefuttatása során létrejövő téves eredmény, amely program- vagy készülékhibából ered.

Hibajavító áramkör (memória) = ECC (ERROR CORRECTION CIRCUIT)

Az ilyen áramköröket integráltan tartalmazó memóriamodul nem csak felismeri, hanem automatikusan ki is javítja a kisebb hibákat, például az 1 bites paritáshibát. Ez hatékony hibavédelmet biztosít, de megdrágítja a memóriát.

Hidegindítás (COLD BOOT, COLD START)

A számítógép bekapcsolásával vagy a Reset gomb megnyomásával kezdődő indítási folyamat. Hidegindításkor a BIOS bekapcsolási öntesztje (POST) teljeskörűen lefut, majd lemezzel a memóriába töltődik az operációs rendszer.

Hierarchikus védelmi rendszer

Az operációs rendszer és felhasználói feladatok (taszkok) védelme privilegizálási szintekkel.

Minden taszk egy védelmi (privilegizálási) szintnek megfelelő privilegizálási osztályba kerül besorolásra. Intel és vele kompatibilis processzorok esetében ez négy szintű:

0-s szinten fut az operációs rendszer magja (kernel)

1-2 szinten futnak az operációs rendszer kiszolgáló rutinjai

3-as szinten futnak a felhasználói feladatok (taszkok).

Home/End (billentyű)

Valamilyen egység (pl. sor, képernyő, lista) elejére illetve végére helyezi a kurzort.

Hordozható szoftver= Platformfüggetlen(PORTABLE)

Több fajta számítógéprendszeren (hardverkonfiguráció vagy operációs rendszer) alatt is futtatható szoftver jelzője. A nagymértékben hordozható szoftver könnyen áthelyezhető más rendszerekre.

Hordozható számítógép(LAPTOP)

Kisméretű számítógép, amelyet hordozható kivitelben gyártanak. A képernyőt magába foglaló fedél a teljes méretűnél kisebb billentyűzetre lecsukható, ezek alkotják a házat, a védőburkolatot is. Egy belső tápegység biztosítja az elektromos hálózattól független, néhány órás használatot.

Horizontális mikroutasítás

Olyan mikroutasítás, amely 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.

A kódolt horizontális mikroutasítás egy kódtáblázat egy sorának címét tartalmazza, ahol a vezérlőbitek megtalálhatók.

Host (hálózatokban)

L.: Nagyszámítógép

Host adapter

L.: Csatoló a processzorhoz

Hozzáférési idő (mágneselem)(DISC ACCESS TIME)

Egy szektor kiolvasásához szükséges átlagos idő. Hozzáférési idő = Pozicionálási idő átlaga + Forgási idő átlaga + Adatátviteli idő + Vezérlési idő

Hub = Kapcsoló, csomóponti csatlakozó

Több, hálózatban működő berendezést (számítógépet, munkaállomást stb.) összekapcsoló eszköz.

Hálózati csatló = LAN-adapter (NETWORK ADAPTER, LAN ADAPTER)

A számítógépet a helyi hálózattal összekapcsoló bővítőkártya vagy alaplagra integrált áramkör.

Hálózati kártya(NETWORK CARD)

A számítógépet a helyi hálózatra csatlakoztató bővítőkártya vagy alaplagra integrált áramkör.

Hálózati munkaállomás (NETWORK WORKSTATION)

Egy mikroszámítógép, amely hálózati ügyfélként (kliens) működik.

Háttértár= Háttértároló, külső tár

(AUXILIARY STORE, BACKING STORE, SECONDARY STORE, BACKING STORAGE, EXTERNAL STORE)

A számítógéphez csatlakoztatott, adatok hosszabb ideig történő tárolására alkalmas berendezés (mágneslemez-tár, mágnesszalag-tár).

Rendszerint azokat a programrészeket, adatokat és az operációs rendszer olyan részeit tárolja, amelyeknek a pillanatnyi feldolgozásban nincs szerepük. A háttértár kapacitása jóval nagyobb, mint a főtáré, de a hozzáférési idő is nagyobb.

Háttértároló-vezérlés

A háttértárvezérlés feladata: írás, olvasás, pozicionálás stb. műveletek irányítása. A vezérlőegység elhelyezkedhet például a merevlemezen egy vezérlőkártyán vagy az alaplapon (ez a korszerű).

Háttértároló

L.: Háttértár

I/O adatátvitel= Input/Output adatátvitel

Adatok kiírása a főtárból az I/O eszközre, vagy beolvasása egy I/O eszköztől a főtárba.

Típusai:

- Programozott I/O
- Megszakításos I/O
- Közvetlen memória-hozzáféréssel (DMA) történő I/O
- I/O processzor alkalmazásával végrehajtott I/O.

I/O busz= I/O sín

Az I/O eszközök (lemezmeghajtók, nyomtató stb.) vezérlése és csatolói, valamint a központi egység többi része (processzor, memória) közötti kommunikációt biztosító busz. Napjainkban használatos típusa a PCI busz.

I/O címtartomány

Az a címtartomány, amelyet az adott I/O eszköz használ a működés során. Az I/O címtartományt az operációs rendszer védi, és így az I/O eszközökkel a felhasználói programok az operációs rendszer közbeiktatásával kommunikálnak.

I/O eszköz

L.: Beviteli/kiviteli eszköz

I/O eszközvezérlő

Egy I/O eszköz vezérlését ellátó hardveregység, amely a mikroszámítógép sínrendszeréhez kapcsolódik. (Sok esetben az I/O eszközvezérlő áramköröket az alaplagra csatlakoztatott kártyák tartalmazzák.) Minden egyes eszközvezérlő

funkcionálisan legalább a következő típusú átmeneti tárolókat tartalmazza:

- Parancsregiszter
- Állapotregiszter
- Adatkiírás illetve beolvasás pufferregiszterei.

I/O eszközök saját sínrendszere

A külső sínrendszer egyik fajtája, például a SCSI lemezcsatoló busza.

I/O interfész = Perifériainterfész

A perifériák (például monitor, billentyűzet stb.) csatlakoztatására szolgáló interfész.

I/O megszakítás

Egy I/O eszköz által kezdeményezett megszakítás. A perifériák és háttértárak a számítógép megszakítási rendszerének segítségével informálják az operációs rendszert az I/O eseményekről. (Például befejeződött egy lemezblokk kiírása.)

I/O processzor = bemeneti/kimeneti processzor (I/O PROCESSOR, INPUT/OUTPUT PROCESSOR)

I/O műveletek önálló vezérlésére alkalmas processzor a nagyszámítógépek csatorna-rendszerében. Utasítások, parancsok küldésével "felprogramozható", ezután teljes önállósággal vezérli az egyes I/O részfeladatokat végrehajtását.

I/O rendszer

A számítógép input/output műveleteket végrehajtó alrendszere. Ez a fogalom átfogja a számítógép megszakítási rendszerét, a DMA-t, az I/O buszrendszert, az alaplapra integrált vezérléseket és csatolókat, az I/O kártyákat, háttértárakat és a perifériákat.

I/O sín

L.: I/O busz

ICH (Input-Output Controller Hub)

Az alaplapi vezérlőáramkör-készlet (chipset) részegysége, amely felelős az I/O sínrendszerek vezérléséért, tartalmazza néhány csatoló vezérlőáramkörét (például IDE, soros port stb.). Ellátja még a CMOS és az RTC vezérlését.

IDE (Integrated Device Electronics) = Integrált eszközelektronika, ATA, AT Attac

Merevlemez-meghajtók régebbi szabványa, ahol a vezérlőelektronika a meghajtón kerül elhelyezésre.

IDT (Interrupt Descriptor Table)

L.: Megszakítás-deszkriptortábla

IEEE (Institute of Electrical and Electronics Engineers)

Az Amerikai Villamos- és Elektromérnökök Intézetének rövidítése.

ILP processzor

L.: Utasításszinten párhuzamos működésű processzor

INTEL ICOMP INDEX

Az INTEL cég teljesítménymérő programja.

A teljesítményindexet a processzoreladások piaci szegmenseinek arányában súlyozva számítják ki. Ezért az index nem használható egy konkrét számítógép-felhasználással összefüggő beruházási döntésnél.

L.: Benchmark

IRQ (Interrupt Request) = Megszakítási kérés

A megszakítás-vezérlő megfelelő vezérlővonalán a jelszint beállítása, amellyel egy I/O eszköz jelzi, hogy megszakítás kiszolgálást kér.

IRQ kiosztás

L.: IRQ

ISA busz (Industry Standard Architecture bus) = Ipari Szabványfelelő busza

Az ISA szabványt (Industry Standard Architecture) az Intel 80286-os processzorával működő PC/AT 16 bites adatbuszát alapul véve 1982-ben alkották meg.

A specifikáció egy 16 bites összeköttetést határoz meg, 8 MHz órajellel, aszinkron működéssel és multiplexált cím- és adatátvitellel (egy sínciklus viszi át a címet, s a következő a 16 bites adatcsomagot).

A multiplexált működés miatt az ISA 16 Mbájt/sec elméleti adatátviteli sebessége a felére csökken, a gyakorlatban a kb. 2-5 Mbájt/sec átviteli teljesítmény is alig érhető el.

A PC-kben az ISA buszt kétféle változatban találhatjuk meg:

- belső ISA busz, amely egy egyszerű interfészt jelent a billentyűzet, a soros és párhuzamos port valamint a floppyegység számára;
- külső ISA bővítő busz, amelynek kártyahelyeihez 16 bites ISA kártyák csatlakoztathatók. (Napjainkban esetenként még 16 bites Sound Blaster kompatibilis hangkártyák csatlakoztatására használják.)

ISA kártya

Az ISA busz kártyahelyeihez csatlakoztatható 16 bites működésű kártya (például Sound Blaster kompatibilis hangkártya).

ISA szabvány

L.: ISA busz

ISA-PCI bridge

L.: PCI-ISA bridge

ISDN(Integrated Services Digital Network)

Integrált szolgáltatású digitális hálózat, amely digitális formában képes hangot, képet, mozgóképet és adatokat továbbítani.

Időosztásos buszhasználat

Buszarbitráció egyik módja. Alkalmazása esetén minden aktív master eszköz meghatározott időszelvre megkapja a buszhasználat jogát (statikus módszer). Csak akkor hatékony, ha a masterek adatátviteli igénye kb. azonos.

Időosztásos multiprogramozás (TIME-SHARING MULTIPROGRAMMING)

A multiprogramozott üzemmód egyik változata, amelynél a programfolyamatokhoz időszeltek kerülnek hozzárendelésre. Ez határozza meg, hogy az adott feladat (taszk)

mennyi időre veheti igénybe a processzort.
Ezen az elven működnek az összes korszerű operációs rendszerek (LINUX, WINDOWS 98 és NT stb.).

Increment

L.: Inkrementálás

Indexelt címzés

Címzési mód, melynél a tényleges címet úgy kapjuk meg, hogy az utasításban lévő címhez (ez például egy vektortáblázat kezdőcíme a memóriában) hozzáadjuk az indexregiszterben lévő értéket.

Informatika

Informatikának nevezzük az információs rendszerek fejlesztésének, működtetésének, hasznosításának törvényszerűségével foglalkozó tudományágat (szakmát).

Információ (INFORMATION)

Új ismeretként értelmezett adat. Az információ definíciójával összefüggésben lényeges hangsúlyozni, hogy az információ mindig kötődik az információt befogadó emberhez.

Információs rendszer (INFORMATION SYSTEM)

Mindazon módszerek, eljárások, folyamatok és eszközök szervezett együttese, mellyel a szervezet tevékenységéhez információt állít elő, befogad, tárol, feldolgoz és továbbít.

Információtechnológia

Magába foglalja mindazon módszereket és eszközöket, melyek az információ előállítását, feldolgozását és továbbítását szolgálják. Ebben az értelemben információtechnológiai eszközöknek minősülnek:

- a hírközlő, kommunikációs és média eszközök,
- a számítástechnikai eszközök,
- az irodatechnikai eszközök (szervező eszközök, mikrofilm, másolók, iratmegsemmisítők stb.).

Infravörös port = SIR (STANDARD INFRARED)

A számítógép infravörös fénysugárral való adatátvitelre szolgáló portja.

Initator

A SCSI csatoló sínrendszerén kommunikációt (sínfoglalást) kezdeményező eszköz. A kezdeményező eszköz adja ki az utasításokat, és vezérli a sín működését. Egy készülék szerepe nem rögzített előre, általában minden eszköz képes az initator és a target szerepét is ellátni.

Inkrementálás = Növelés (INCREMENT)

Egy szám értékének pozitív irányú változtatása.
Ha eltérő értelmű külön utasítás nincs rá, általában eggyel való növelést jelent.

Insert (billentyű)

Szövegszerkesztő programokban a beszúrás (Insert) vagy felülírás (Overwrite) üzemmód között lehet választani. Beszúrásnál az új karakter a kurzor pozíciójába kerül és a szöveg további része jobbra tolódik, felülírásnál a kurzor "meletti" karakter kicserélődik a leütöttel.

Instruction pointer

L.: Utasítás-számláló regiszter

Instruction register

L.: Utasítás-regiszter

Intel

Integrált áramköri alkatrészeket gyártó cég, amely elsőként fejlesztett ki kereskedelmi forgalomban is kapható mikroprocesszort (a 4004-et). Az Intel terméke a 80x86-os és ennek továbbfejlesztése, a Pentium processzorcsalád.

Interactiv processing

L.: Interaktív feldolgozás

Interaktív feldolgozás (INTERACTIV PROCESSING)

Olyan számítógéprendszer üzemmód, amelynél a felhasználó állandó és folyamatos kapcsolatot tart a feladatot végrehajtó programfolyamattal. A felhasználó a feladat végrehajtását maga vezérli, szükség esetén lépésenként módosítja a részfeladatok sorrendjét.

Két altípusa van attól függően, hogy a felhasználó az ember vagy egy gépi folyamat:

- dialógus üzemmód
- folyamatvezérelt üzemmód.

Interface

L.: Interfész

Interfész= Illesztőegység, csatoló, adapter(INTERFACE)

1. Két - akár hardver, akár szoftver - funkcionális egység együttműködését biztosító előírások összessége. Ezen előírások kiterjednek többek között - hardver esetén - a fizikai-mechanikai jellemzőkre (például csatlakozók), a definiált jelekre, azok elektromos jellemzőire, az egyes funkciókat realizáló jelszekvenciákra, valamint - szoftvernél is - a definiált műveletek (például kapcsolatfelvétel, adatátvitel, szétkapcsolás) megvalósítására.

Megjegyzések:

a) Az interfész két funkcionális egység közötti közös határfelületként is értelmezhető, amelyet a fent nevezett előírások határoznak meg.

b) Ebbe a fogalomba beletartozik két különböző funkciójú berendezés összekapcsolásának specifikációja is.

2. Az interfész-specifikációkat (lásd 1. pont) realizáló hardver- vagy szoftvereszköz.

Interfészáramkör

Feladata az adatok továbbítása az I/O eszköz és a memória között, az eszközök vezérlése, az állapotinformációk lekérése, megszakítások generálása állapotváltozásokkor.

Internal cache memory

L.: Elsőszintű cache

Interrupt Descriptor Table

L.: Megszakítás-deszkriptortábla

Interrupt request

L.: Megszakítási kérelem

Interrupt

L.: Megszakítás

Intranet

Olyan helyi hálózat, amely, és az Internettel azonos protokollt használ.

IrDA (Infrared Data Association)= Infravörös adatkapcsolat

A vezeték nélküli kapcsolatok fejlesztése céljából létrehozott szabvány, amely az adatátvitelt az infravörös fényintenzitás modulálásával valósítja meg.

Jelzőbit (FLAG)

Olyan állapotinformáció, mely egy rendszer (hardverrendszer vagy program stb.) működési módjáról, működési feltételeiről vagy a működés közben előálló állapotokról (pl. eredmény=0, kiviteli/beviteli műveletek eredményes vagy eredménytelen végrehajtása stb.) ad felvilágosítást. Különbséget kell tenni hardver úton vagy szoftver úton (programváltozóként, illetve logikai változóként) megvalósított jelzőbitek között. A jelzőbit beállíthatók, lekérdezhetők, illetve módosíthatók.

Kapu (tárolóvédelem) (GATE)

A különböző privilégium szintű programfolyamatok közötti kapcsolatok és átmenetek (kód- és adatszegmensekhez való hozzáférés, taszkváltás) ellenőrzését szolgáló struktúra.

Az Intel processzorok négy fajta kaput kezelnek védett üzemmódban:

- Callkapu
- Megszakításkapu
- Trapkapu
- Taszkkapu

Karakter (CHARACTER)

Az írott vagy nyomtatott szöveg egységelemét nevezzük karakternek.

Katalógus (könyvtár, directory)

Olyan speciális állomány, mely az adatokat tartalmazó állományok jellemzőit tárolja, lehetővé teszi csoportosításukat.

Kazetta= Mágnesszalag kazetta(CARTRIDGE, CASSETTE)

Adat vagy program tárolására alkalmas mágnesszalagot tartalmazó kazetta.

Kibocsátási politika (szuperskalár processzorok)

A szuperskalár utasítás-kibocsátásnál az adat- és vezérlésfüggőségek kezelésének eljárása.

Az adat- és vezérlésfüggőségek okozta blokkolódásokat a processzor kezelheti az utasítás-kibocsátás blokkolásával, vagy a függő utasítás pufferelemével a végrehajtóegység előtt (várakoztatás).

Kibocsátási ráta

A szuperskalár utasítás-kibocsátás egyik összetevője, amely megadja a processzor által ciklusonként kibocsátható, vagyis a végrehajtó egységekhez továbbítható utasítások

számát.

Kisszámítógép

L.: Mikroszámítógép

Kiszolgáló feladat átbocsátó képessége

A kiszolgáló (szerver) által időegység alatt végrehajtott feladatok száma.
Nagy átbocsátóképességet olyan "ideális" hardver-, szoftver-architektúrákkal lehet elérni, amelyek biztosítják, hogy

- a szerver mindig dolgozzon,
- a várakozósor sohase legyen üres.

Kiszolgálóeszköz (SLAVE)

L.: Szolga

Kivétel (EXCEPTION)

Az utasítások szabályszerű feldolgozását megszakító esemény kivétel, ha a programot a programutasítás végrehajtása alatt a processzoron belül fellépő esemény miatt kell megszakítani (például nullával való osztás, laphiba).
Kivétel altípusai: hiba, csapda, abort, programozott kivétel.

Kliens-szerver architektúra = Ügyfél-kiszolgáló kapcsolat, ügyfél-gazda kapcsolata

(CLIENT-SERVER ARCHITECTURE)

Olyan számítógéprendszer modell, amelyben a kliens és a szerver számítógépeket egy hálózat kapcsolja össze, az alkalmazásokat egy előtérben futó ügyfél és egy háttérben futó kiszolgáló (például fájl, adatbázis, kommunikációkezelés) program feladatmegosztása és kommunikációja valósítja meg.

Kliens= Ügyfél(CLIENT)

Olyan számítógép, amely hálózaton keresztül egy másik számítógépen (szerver) futó programok szolgáltatásait használja.

Kombinációs döntési hálózat

Általában olyan digitális hálózat, melynek bármely bemeneti állapotához, jelkombinációjához egy definiált kimeneti jelkombináció tartozik.

Kombinációs áramkör

Olyan logikai áramkör, amely nem tartalmaz tárolóelemet, és amelynek kimeneti értékei csakis a bemeneti értékek pillanatnyi értékétől függenek.

Kommunikáció (COMMUNICATION)

Az információ adása, átvitele és vétele, mely közelítőleg tömegkommunikációra és egyéni kommunikációra osztható. A tömegkommunikáció információk közvetítése a nyilvánosság számára újságok, rádió, televízió stb. útján. Az egyéni kommunikáció két vagy kisszámú résztvevő közötti információtovábbítás pl. a telefon-, telex-, telefax-hálózat útján.

Kommunikációs protokoll

Két eszköz közötti adat és vezérlő információcsere szabályainak összessége.

Kompatibilitás (COMPATIBILITY)

Két eszköz, készülék, berendezés, program együttműködését, egymással való helyettesíthetőségét vagy összekapcsolhatóságát jelentő tulajdonság:

- Két számítógép akkor kompatibilis, ha az egyikre készült programok a másikon minden változtatás nélkül futtathatóak.
- Két eszköz, készülék, berendezés, program akkor kompatibilis, ha az egyik által szolgáltatott információt a másik helyesen értelmezi és dolgozza fel.

Komplex utasításkészletű számítógép = CISC (COMPLEX INSTRUCTION SET COMPUTER)

Olyan számítógép-architektúra, amelynek processzora

- nagyszámú utasítást tartalmazó utasításkészlettel rendelkezik;
- az utasítások szerkezete bonyolult;
- többfajta memóriacímzési módot tesz lehetővé;
- az utasítás-végrehajtás vertikális mikroprogrammal vezérelt.

Konstans (CONSTANT)

Programnyelvben olyan numerikus vagy szöveges adatobjektum melynek értéke egy adott programon belül nem változik. Kifejezésekben operandusként szerepelhet.

Kártya

L.: Bővítőkártya

Kártyahely

Az alaplapon található csatlakozóhely, amely vezérlő és adapter áramkörüi kártyák csatlakoztatására szolgál.

Képernyő (DISPLAY SCREEN)

A monitornak az adatok és képek megjelenítésére szolgáló része.

Képmemória (FRAME BUFFER)

A memóriának az a része, ahol a számítógép a pixelek színkódjaival a képernyő tartalmát tárolja. Általában a monitorvezérlő kártyán elhelyezkedő memóriaterület. Méretét a képernyő felbontása és az alkalmazott színkódolás határozza meg. Fizikailag a monitorvezérlő kártya RAM tárolójában helyezkedik el, feltöltését a kártya grafikus processzora vezérli.

Képpont (PIXEL, PICTURE ELEMENT)

A képernyő legkisebb olyan egysége, amelyet a számítógép kezelni tud. Színes monitor esetében a három RGB alapszínnek megfelelő elemi pontokból áll.

Kétszeres kézfogás (sínrendszerek) (FULLY INTERLOCKED HANDSHAKING)

Aszinkron sínvezérlésnél a kommunikáló eszközök összehangolt működését biztosító eljárás.

Jellemzője, hogy az adatátvitel következő lépésére csak akkor lehet rátérni, ha az adatátvitel előző lépése ellenőrzötten (visszaigazoltan), hibátlanul befejeződött.

Kézfogás (HANDSHAKING)

Aszinkron adatátviteli összeköttetés szervezett és hibamentes lefutását biztosító eljárás (Handshake protokoll).

Legegyszerűbb esetben két egység úgy működik együtt a kézfogás elve alapján, hogy az az egység, amely valamilyen tevékenységet kezdeményez, egy kérést (request)

bocsát ki a másik egység felé. Amikor a kért egység kész a kért tevékenység elvégzésére, egy nyugtázó jelet (acknowledgement) küld vissza a kérő egységnek.

Kódkonverzió (CODE CONVERSION)

Adat egyik ábrázolási formáról a másikra való átalakítása, például bináris szám átalakítása binárisan kócolt decimális számra.

Kódkonverzió(CODE CONVERSION)

Adat egyik ábrázolási formáról a másikra való átalakítása, például bináris szám átalakítása binárisan kócolt decimális számra.

Kódolás (CODE)

Az egyes hírekben és közleményekben lévő adatok formai átalakítását kódolásnak nevezzük.

A kódolás tehát áttérést jelent egy jelkészlet és ezzel összefüggő szabályrendszer használatáról egy másik jelkészletre és szabályrendszerre.

Kódrendszer

A kódolás során használt jelkészletet és formai szabályrendszert együttesen kódrendszernek nevezzük.

Körbejáró eljárás (sínrendszerek)

A párhuzamos sínfoglalás esetében a master eszközök prioritását meghatározó egyik eljárás.

Minden sínhasználatot követően a korábban legalacsonyabb prioritású eszköz kapja meg a legmagasabb prioritást, az összes többi eggyel alacsonyabb prioritási szintre kerül.

Köteget feldolgozás (BATCH PROCESSING)

Előre összegyűjtött adatok feldolgozása vagy előre összegyűjtött munkák elvégzése oly módon, hogy a folyamat nem igényli a felhasználó beavatkozását. A köteget feldolgozással csökkenthetők az állásidők, mivel egy munka befejezése után a rendszer automatikusan megkezdí a következő munka feldolgozását. Előfeltétele többek között egy operációs rendszer megléte, illetve egy munkavezérlő nyelv (job control language) használata, amelynek segítségével a felhasználó specifikálni tudja az elvégzendő tevékenységeket.

Kötet (meghajtó, volume, drive)

Az a névvel ellátott logikai egység, mely segítségével az operációs rendszer a számítógép háttértárolóit kezeli.

Közepes számítógép (MIDRANGE COMPUTER)

Közepes méretű számítógép. A kifejezés azonos a miniszámítógép szó jelentésével, azzal a kivétellel, hogy az egyfelhasználós munkaállomások nem tartoznak a közepes számítógépek közé.

(Megjegyzés: A magyar terminológia ezt a kategóriát nem használja.)

Központi egység= CPU(CENTRAL PROCESSING UNIT)

A szakirodalom nem egységes a központi egység fogalom értelmezésében.

1. Az egyik felfogás szerint a processzor a számítógép központi egysége, amely az utasítások interpretálását és végrehajtását vezérlő áramköröket tartalmazza.

2. A másik felfogás esetében a központi egység fogalomba tartozik a processzor, az operatív tár (belső tár) és a beviteli/kiviteli egység vezérlői, lényegében minden, ami a számítógépházban található.

Központosított busz arbitráció

Sínhasználat jogának eldöntésére szolgáló folyamat egyik típusa.

A buszhasználatot igénylő mastelek egy "request" jellel jelzik az igényeiket egy központi arbiternek, amely az arbitrációs algoritmus szerint kiválaszt egy mastelet, és az igény elfogadását a "grant" jellel igazolja vissza.

Ennek három módszere terjedt el:

- soros sínfoglalási eljárás
- párhuzamos sínfoglalási eljárás
- az előző két módszer kombinált alkalmazása.

Közvetett I/O eszközcímzés (MEMORY MAPPED ADDRESSING)

A sínrendszerhez csatlakozó I/O eszközezőrlők címzésének olyan módja, amikor a címzés úgy történik, mintha az I/O eszközezőrlő tárolója a főtár része lenne. (Így címzik például a monitorvezőrlő kártya grafikus memóriát.)

Közvetett I/O utasítás

A főtárra vonatkozó programutasításokkal (például MOVE) kezelhetők az I/O eszközezőrlők tárolói, az I/O eszközezőrlők programozásához a processzor utasításkészletében nincs elkülönült I/O utasítás.

Közvetett bemenet

L.: Offline

Közvetett címzés

A közvetett cím egy olyan memóriaterületet címez meg, amely az adat tényleges címét tartalmazza.

A közvetett címet sokszor pointernek (mutatónak) is szokás nevezni.

Közvetlen I/O eszközcímzés

Az I/O eszközezőrlőkhöz tartozó tárolók címzésének olyan módja, amikor az eszközezőrlő minden tárolójának (pufferek, regiszterek) saját, önálló címe van.

Közvetlen adatszímzés

Az utasításban maga az az adat található meg, amellyel a műveletet végre kell hajtani. Ezt a címzési eljárást még közvetlen értékadó címzésnek, álcímzésnek és literálcímzésnek is szokták nevezni.

Közvetlen bemenet

L.: On-line

Közvetlen hozzáférésű tároló

Olyan tároló, melyben az adatokat felvitelük sorrendjétől függetlenül lehet elérni, bármelyik tárolt adat beírása körülbelül azonos idő alatt történik meg, valamennyi rekesz hozzáférési ideje lényegében egyenlő.

Közvetlen hozzáférésű eszköz (DIRECT-ACCESS DEVICE)

Olyan háttértároló, amely lehetővé teszi a keresett adatot tartalmazó memóriablokk közvetlen megcímezését, és ezáltal az adatoknak a teljes tároló végigolvasása nélkül történő kiolvasását vagy beírását.

Legfontosabb közvetlen elérésű háttértároló a mágneslemez.

Közvetlen leképezésű cache (DIRECT MAPPING CACHE)

Olyan cachetároló, amelynek sorait a főtárcím bitjeinek egy meghatározott részével, mint sorindexszel címezzük.

A közvetlen leképezésű cacheek esetében ezért a különböző blokkokban levő, de azonos sorindexű blokkokban található adatok hozzáférése rendkívül lelassul. (Ez minden esetben blokkcserét eredményez a cacheben.)

Ugyanakkor ez a cachetároló olcsó és gyors visszakeresést biztosít.

Közvetlen memória-hozzáférés= DMA(DIRECT MEMORY ACCESS)

A processzor egy I/O művelet végrehajtásához szükséges információkat átadja egy, a processzortól független DMA-vezérlőnek, amely ezt követően az adatátvitelt a memória és az I/O eszköz között önállóan irányítja. Ezáltal a processzor felszabadul más feladatok végrehajtására.

Közvetlen átírás (WRITE THROUGH)

A cacheben megváltoztatott adatok főtárba történő visszaírásának egyik eljárása, amikor a gyorsítótár írásával együtt azonnal megtörténik a főtár írása is. Ennek alkalmazása esetén a főtár és a cache adategyezősége "automatikusan" biztosított, de a főtár írásműveleteit a cache nem gyorsítja.

A közvetlen átírás módszerének hatékonyságát javítja puffertelt közvetlen átírás.

Külső buszrendszer

A számítógép sínrendszerek egyik típusa, amely a processzort köti össze a számítógép különböző reszegységeivel.

Sebessége és az összekapcsolt eszközök alapján lehet: helyi sín, rendszersín, I/O eszközök saját sine és a számítógéprendszerek közötti busz.

Külső gyorsító tár

L.: Második szintű cache

L1 cache

L.: Elsőszintű cache

L2 cache

L.: Második szintű cache

LAN (Local Area Network)= Helyi hálózat

Kis földrajzi kiterjedésű (mint néhány száz méter vagy néhány kilométer átmérőjű körön belül elhelyezkedő) számítógépes (kommunikációs és szolgáltató) hálózat. Szolgáltatásai a hálózatba kapcsolt munkaállomásokon (terminálokon) keresztül vehetők igénybe, amelyek általában személyi számítógépek.

LAN-adapter

L.: Hálózati csatoló

LDCM (Local Area Network Desk Manager)

Olyan program, amely (alaplaptól függően) lehetővé teszi a rendszergazda számára a hálózatba kötött számítógépek működésének távoli felügyeletét.

LDT

L.: Lokális deszkriptortábla

LPT (Line-Printer Terminal)

A párhuzamos port szimbolikus neve.

LRU stratégia

L.: Legrégebben használt blokk stratégiája (cache)

LRU

L.: Legrégebben használt elvű algoritmus

Lap (PAGE)

A virtuális tár rögzített méretű, nem átlapolható blokkjai.

Laphiba (PAGE FAULT)

Egy programfolyamat egy utasítása olyan virtuális címre hivatkozik, amelynek megfelelő lap nincs a főtárban.

A laphiba kivétel megszakítja a futó programot, és a vezérlést megkapja az operációs rendszer laphiba kezelő rutinja, amely a hivatkozott lapot betölti a főtárba.

Lapkakészlet

L.: Csipkészlet

Lapkeret (PAGE FRAME)

Lapozást alkalmazó virtuális tárkezelés esetében a főtárnak a lapmérettel megegyező nagyságú része. Például 4 Kbájtos lapok esetében a 0-ás lapkeret a 0-ás címtől a 4095-ig tartó főtárrész.

Lapozás (PAGING)

A virtuális tár kezelésének egyik módja. Alkalmazásánál a virtuális memóriatartomány rögzített méretű blokkokra (lapokra) van felosztva.

Laptop

L.: Hordozható számítógép

Laptáblázat (PAGE TABLE)

A központi memóriába beolvasott lapok főtárbeli fizikai kezdőcímét - azaz azoknak a lapkereteknek a címét, ahová a lap elhelyezésre került - tartalmazó memória-táblázat.

Lebegőpontos szám

Három mező alkot tehát egy lebegőpontos számot: az előjel, az exponens és a mantissza.

Tehát a lebegőpontos számok számítógépes ábrázolása során 2 fixpontos számot, a mantisszát, és az exponenst kell együtt kezelnünk.

Lebegőpontos processzor = FPU, társprocesszor, kooprocesszor, numerikus segédprocesszor (FLOATING POINT UNIT, COPROCESSOR, NUMERIC COPROCESSOR)

Olyan mikroprocesszor, amely másodlagos (segéd-) processzorként utasításokat hajt végre. A mai processzorok (az i486-os processzortól kezdve) a társprocesszorokat már beépített egységként tartalmazzák.

Lebegőpontos társprocesszor

L.: Lebegőpontos processzor

Lebegőpontos végrehajtóegység (FLOATING POINT UNIT)

A lebegőpontos számításokat végző áramköri egység.

A 486DX processzoroktól kezdve a lebegőpontos egységet a processzor tartalmazza.

Lefelé való kompatibilitás = Felülről való kompatibilitás (DOWNWARD COMPATIBILITY)

Egy fejlettebb rendszernek az a képessége, hogy a nála kevésbé fejlett rendszeren futtatható programokat végrehajtja.

Legrégebben használt blokk stratégiája (cache) = LRU stratégia (LAST RECENTLY USED)

A legelterjedtebb blokkhelyettesítési algoritmus, amelynél az új főtárblokk cachebe történő bemásolásakor a processzor által legrégebben használt főtárblokk adatai kerülnek felülírásra.

Legrégebben használt elvű algoritmus = LRU (LAST RECENTLY USED)

Ha a tárhierarchiában egy gyorsabb elérésű tároló egy blokkját egy lassú elérésű tárblokkal kell helyettesíteni, akkor a processzor által legrégebben használt blokk kerül kiválasztásra (felülírásra).

Legrégebben használt eszköz eljárás (sínfoglalás)

A párhuzamos sínfoglalás esetében a master eszközök prioritását meghatározó egyik eljárás.

Alkalmazásakor az a master eszköz kapja meg a sínhasználat jogát, amely a sít legrégebben vette igénybe.

Lemezgyorsítás = Merevlemez-gyorsítás

A főtár és a mágneslemez közötti adatátvitel gyorsítása cachetár alkalmazásával. A lemezgyorsítás megvalósítható szoftveres vagy hardveres úton.

Lemezinterfész (DISC INTERFACE)

1. A lemezmeghajtót a számítógép-rendszerhez kapcsoló áramkör.
2. A lemezmeghajtóknak a számítógép-rendszerhez való csatlakoztatására vonatkozó szabvány. (Például ST506 szabvány egyike a lemezekre vonatkozó interfészszabványoknak.)

Lemezmeghajtó (DISK DRIVE, FLOPPY DISK DRIVE)

Elektromechanikus eszköz, amely olvas egy mágneses lemezről, vagy ír oda. A lemezmeghajtó főbb részei közé tartoznak a tengely, amelyre a lemezt szerelik, egy meghajtómotor, amely forgatja a lemezt a meghajtó működése közben, egy vagy több író-olvasó fej, egy másik motor, amely a lemez fölé pozicionálja az író-olvasó fejeket és egy vezérlőáramkör, amely szinkronizálja az olvasó/író tevékenységeket, és irányítja az adatcserét a számítógép központi egységével. Általában két féle lemezmeghajtót használnak: a hajlékonylemez-meghajtókat és a merevlemez-meghajtókat.

Lemezvezérlő(DISC CONTROLLER)

Speciális célú lapka és a hozzátartozó áramkör, amely vezérli a számítógép lemezegységére történő írást, illetve az onnan való olvasást. A lemezvezérlőket hajlékonylemez-meghajtóknál és merevlemezekenél használják.

Linux

PC-ken szabadon terjeszthető és felhasználható UNIX operációs rendszer.

Local Area Network

L.: LAN

Local Descriptor Table

L.: Lokális deszkriptortábla

Local bus

L.: Helyi sín

Logikai áramkör

Olyan áramkör amely diszkrét értékű bemenő jelekből, meghatározott matematikai logikai függvényeknek megfelelő kimenőjelet állít elő. Ezen logikai értékeket az áramkör valamely elektromos jellemzőjéhez rendeljük hozzá.

Lokális deszkriptortábla = LDT (LOCAL DESCRIPTOR TABLE)

Egy taszkhoz tartozó deszkriptorokat tartalmazó memóriatáblázat, amelyet az operációs rendszer minden feladathoz létrehoz.

Lézernyomtató (LASER PRINTER)

Egy teljes oldal egy lépésben történő kinyomtatására alkalmas eszköz, amely a nyomtatási kép kialakításához lézersugarat alkalmaz.

MCH (Memory Controller Hub)

A korszerű alaplapi vezérlőáramkör-készlet (chipset) részegysége. Ez a memória-vezérlő csomópont felelős a processzor és az L2 cache, a rendszermemória és az AGP közötti adatáramlás irányításáért.

MESI protokoll (MODIFIED, EXCLUSIVE, SHARED, INVALID PROTOCOL)

A főtár és a cachetárak azonosságát biztosító szabványeljárás.

Alkalmazása esetén a cachetárak blokkjainak lehetséges állapotai:

- módosított, amely esetben a cachetár blokkja a főtárblokkhoz képest módosítva lett, és most ez tartalmazza az aktuális adatokat;
- kizárólagos, ekkor a cacheblokk megegyezik a főtárblokkal, és ez a blokk más cacheben nem található meg;
- megosztott vagy közös, ekkor a cacheblokk a főtárral egyező, érvényes adatokat tartalmaz, de ez a blokk több cacheben is megtalálható;
- érvénytelen, amikor a cacheblokk már nem aktuális adatokat tartalmaz (például egy másik cacheben már módosították).

MFLOPS (Millions of Floating Point Operations per Second)

= millió lebegőpontos utaitás másodpercenként

A számítógépek lebegőpontos teljesítményét kifejező mutató, az időegység alatt

végrehajtott lebegőpontos utasítások átlagos száma, azaz a millió lebegőpontos művelet/másodperc.

MIMD architektúrájú számítógép (MULTIPLE INSTRUCTION STREAM MULTIPLE DATA STREAM)

Olyan számítógép, amely több utasításfolyammal több adatfolyamatot dolgoz fel. Ebbe a kategóriába tartoznak a multiprocesszoros számítógépek (több vezérlőegységgel) és a nem soros utasítás-végrehajtás elvén működő számítógépek (adatvezérelt számítógépek, neuronális hálók).

MIPS = millió utasítás másodpercenként (Million Instructions Per Second)

A számítógép számítási sebességének mértékegysége. Pontosabb értelmezésben meg kell adni, hogy megadása milyen utasításmixre történt.

MISD architektúrájú számítógép (MULTIPLE INSTRUCTION STREAM SINGLE DATA STREAM)

Olyan számítógép, amely több utasításfolyammal egyetlen adatfolyamatot dolgoz fel. Egyes szakértők ide sorolják a pipeline (futószalag) szervezésű processzorokat, illetve a hibatűrő architektúrákat. (Utóbbiak többszörösen végzik el a műveleteket azonos adatokon, és az eredményt hibavédelmi célból összehasonlítják.)

MISS javítási idő

A cache teljesítményének egyik legfontosabb mérőszáma. Meghatározza, hogy egy adat átlagosan mennyi idő alatt adható át a processzornak, ha az adatot tartalmazó blokk nem található meg a cacheben. Értéke: a hit/miss meghatározásának ideje + adatblokk kiolvasás a főtárból + adat átadási ideje a processzornak.

MISS-RATE

Az adatoknak a cacheben történő visszakeresése során a tévesztések (miss) százalékos aránya.

MMX (Multimedia Extension)

Az 1997. elején az Intel által bevezetett multimédiás utasításkészlet-kiegészítés, amelyet minden azóta megjelent Intel vagy azzal kompatibilis processzor is alkalmaz.

MMX processzor

A multimédiás képfeldolgozást gyorsító utasításkészlettel rendelkező processzor, amely képes az ún. MMX adattípusok feldolgozására vektorszámítógépként (SIMD).

MODEM (MODulator/DEModulator) = Modulátor/demodulátor

Olyan eszköz, amely egy (digitális) bitsorozatot analóg átviteli csatornán történő átvitelre alkalmas analóg jellé alakít át (modulál), illetve a vett analóg jeleket (digitális) bitsorozattá alakítja vissza (demodulál).

MOPS = millió utasítás másodpercenként (Million Operations Per Second)

A számítógép számítási sebességének mértékegysége. Pontosabb értelmezésben meg kell adni, hogy megadása milyen utasításmixre történt.

MOV (Adatmozgató utasítás)

Adatmozgatás memória és regiszter, regiszter és regiszter között. Az utasítás működése: A céloperandus felveszi a forrásoperandus értékét, a forrásoperandus változatlan marad (ha a kettő nem azonos). Az utasítás végrehajtása a flag-eket nem állítja be.

Magasszintű programnyelv

Az emberi nyelvhez közelítő programnyelv. (FORTRAN 1954, ALGOL 60, COBOL 1959, BASIC 1964, PASCAL 1968, C 1974), melyekkel könnyebben lehet gyorsan áttekinthető és hibamentes programokat írni. Természetesen ennek az ára az volt – mivel a magasszintű programnyelveknél egy utasítás általában már több gépi kódú utasításra lesz lefordítva –, hogy a fordítóprogramok is egyre bonyolultabbá váltak.

Mainframe

L.: Nagyszámítógép

Master

A sín lefoglalására jogosult eszköz. Elindít és befejez egy tranzakciót, az adatátvitelhez címet küld.

Maszkolható megszakítási kérelem (MASCABLE INTERRUPT)

Olyan megszakítás-kérelem, amelynek kiszolgálása átmenetileg letiltható. Engedélyezése vagy tiltása a megszakítás-vezérlő egy regiszterében egy megfelelő bit beállításával történik.

Md (DOS parancs)

Olyan könyvtár létrehozására szolgáló parancs, mely az aktuális tartózkodási hely könyvtárából fog nyílni.

Megszakítás (INTERRUPT)

A számítógépi program végrehajtásának valamely esemény által előidézett felfüggesztése oly módon, hogy a program folytathatósága megmaradjon. A programmegszakítások lehetséges okai: 1. óramegszakítás, más programra való áttérés céljából (idoosztásos rendszereknél); 2. perifériális berendezés kiszolgáláskérése; 3. rendszerhívási utasítás (SVC, system call) 4. az operátor beavatkozása; 5. túlcsoordulás összeadásnál, osztásnál; 6: tápfeszültség-kimaradás stb.

Megszakítás-deszkriptortábla = IDT (INTERRUPT DESCRIPTOR TABLE)

A megszakítások és kivételek kezeléséhez szükséges deszkriptorokat tartalmazó memóriatáblázat.

Védett üzemmódban a megszakításokhoz és a kivételekhez tartozó kapuk (megszakítás, trap, taszk) deszkriptorait tartalmazza.

Megszakítás-kiszolgálás lépései

Az IBM-PC kompatibilis számítógépek vektoros megszakítás-kérelmének kiszolgálási lépései:

A hardver által:

- a megszakítás-vezérlő beállítja a megszakításkérő vezérlő vonal jelszintjét, ezzel jelzi a processzornak a megszakítás-kérelmet (INT jel);
- a processzor visszaigazolja a megszakítás-kérelem elfogadását (IACK jel);
- ezt követően a vezérlő a sínre küldi a megszakítási vektor elemének sorszámát;
- a processzor tárolja a megszakítási vektor elemének sorszámát;
- a processzor elmenti a verembe az utasítás-számláló és az állapotregiszter tartalmát;
- a processzor a megszakítási vektor elemszámára alapján a megszakítás-kiszolgáló rutin kezdőcímét betölti az utasítás-számláló regiszterbe, és ezzel megkezdődik a megszakítás-kiszolgáló rutin végrehajtása.

Az operációs rendszer által:

- a megszakított program adatainak elmentése verembe (ha szükséges);
- a megszakítás okának behatárolása;
- a kiszolgáláshoz szükséges adatok összegyűjtése;
- a megszakítást okozó esemény kezelése;
- a megszakított program adatainak visszatöltése;
- a megszakítás-kiszolgáló rutin befejezésének jelzése.

A hardver által:

- az elmentett állapot és utasítás-számláló regiszter tartalmának visszatöltése és a megszakított program folytatása.

Megszakítás-vezérlő egység (INTERRUPT CONTROL UNIT)

A megszakítás kiszolgálásához szükséges legfontosabb hardverfeladatokat végzi el egy Intel processzoros architektúrában:

- fogadja a megszakításkérő vezérlővonalakon (IRQ) a megszakítás-kérelmeket;
- vizsgálja, hogy az igényelt megszakítás nincs-e maszkolással letiltva,
- vizsgálja és értékeli a megszakítás prioritását;
- az INT (Interrupt) vezetéken közli a megszakítás-kérést a processzorral;
- ha az IACK (Interrupt Acknowledgment) vezetéken a processzor visszaigazolja, hogy kész a kérés fogadására, akkor a megszakítás-vezérlő átadja (sínre teszi) a processzornak a megszakításhoz tartozó megszakításvektor elem címét.

Megszakítás

Olyan, a futó programhoz képest külső esemény (például egy periféria jelzi az I/O művelet befejezését) amelynek kezelése a futó program utasítás végrehajtásának átmeneti felfüggesztésével történik.

Megszakítási kiszolgálás (INTERRUPT SERVICING)

Tevékenységsor, amelyet a megszakítási rendszer hajt végre, amikor a megszakítást kiváltó eseményt kezel.

Megszakítási kérelem= IRQ(INTERRUPT REQUEST)

Egy hardvereszköz jelzése egy vezérlő vonal jelszintjének beállításával egy olyan esemény bekövetkezéséről, amelynek kezelése csak az aktuálisan futó program végrehajtásának átmeneti felfüggesztésével lehetséges.

Megszakítási rendszer(INTERRUPT SYSTEM)

A számítógépben a megszakítási kérelmek fogadását, kiszolgálását végző hardver-, szoftvereszközök és eljárások összessége.

Megszakítási vektortáblázat

A megszakításokat kiszolgáló rutinok kezdőcímeit tartalmazó memória-táblázat, amelyre hivatkozó azonosító sorszámot (index) a megszakításvezérlő adja át a processzornak. Az Intel és vele kompatibilis processzoroknál alkalmazott megszakítási eljárás.

Megszakításkapu (INTERRUPT GATE)

Az Intel és vele kompatibilis processzorok tárolóvédelmi rendszerének négy kaputípusa közül az egyik, amelyet a rendszer a hardvermegszakítások, a processzor által detektált kivételek (faults, traps, aborts) és a programvezérelt megszakítások (INT, IRET utasítások) kezeléséhez használ.

Megszakításos I/O

Az I/O műveletek végrehajtásának leggyakoribb módszere.

Az átvitel kezdetén, a megfelelő utasítás kiadásával, a processzor jelzi az I/O eszköz számára az átvitelre vonatkozó igényét. Ha az I/O eszköz az átvitelre alkalmas állapotban van, akkor ezt a processzor felé küldött megszakítási kérelmével jelzi, és az adatátvitel befejezéséről szintén megszakítás-kérellemmel informálja az I/O eszköz a

processzort.

A megszakításos I/O művelet esetén a processzor csak a megszakítás kiszolgálásának időtartama alatt foglalt.

Megszakítással kezdeményezett bemeneti/kimeneti (I/O) adatátvitel

Az adatátvitelt egy programmegszakítás előzi meg. Ekkor egy I/O eszköz jelzésére a processzor félbeszakítja az éppen futó program végrehajtását, majd az I/O eszköz kérését végrehajtja.

Melegindítás= Szoftveres indítás(WARM BOOT, WARM START)

Bekapcsolt számítógép újraindítása anélkül, hogy a főkapcsolót kikapcsolnánk. A melegindítás a billentyűzetről kezdeményezhető, a Ctrl+Alt+Delete billentyűzetkombinációval. Ekkor a gép újraindul, viszont nagyon sok dolog nem törlődik (interrupt vektorok, egyes vírusok stb.), komolyabb lefagyás esetén nem célszerű alkalmazni.

Mem (DOS parancs)

A számítógép memóriájának felosztása, rezidens programok felsorolása.

Memory mapped addressing

L.: Közvetett I/O eszközcímezés

Memória-konzisztencia (szuperskalár processzorok)

Párhuzamos utasítás-feldolgozásnál a soros konzisztencia egyik összetevője, a memória-hozzáféréseknek az eredeti utasítássorrend szerinti végrehajtása. (Mivel szuperskalár processzorok párhuzamos utasítás-feldolgozása esetében az egyes végrehajtóegységek az utasítások eredményeit az eredeti utasítássorrendtől eltérően is előállíthatják, az eredményadatokat átmenetileg tárolni kell, és a programutasításban előírt, végleges helyükre az eredeti utasítássorrendnek megfelelően kell beírni.)

Memória

L.: Tár

Memóriabank

Átlapolt memóriacímzésnél a memória önállóan címezhető, írható és olvasható egysége.

Memóriacsip

A memória alapegységét realizáló integrált áramkör. Memóriakapacitását Mbit-ben adják meg.

Memóriafoglatlat

Alaplapi részegység, amelybe a számítógép központi memóriájának moduljait illesztik.

Memóriamodul (MEMORY MODULE)

Meghatározott, 2-hatvány darabszámú memóriacsipből összeállított memóriaegység, amelyet különálló sokérintkezős nyáklapra szerelnek, és az alaplapi memóriafoglatatába kell behelyezni.

Merevlemez-gyorsítás

L.: Lemezgyorsítás

Merevlemez

L.: Merevlemez tárolás

Merevlemez tárolás = Merevlemez diszk (HARD DISK, RIGID DISK)

Olyan mágneslemez tárolás, amely mágneses réteggel borított kerámia- vagy alumíniumlemezből készül. A merevlemez tárolás nagy tömegű adat tárolására, és az adatok közvetlen, viszonylag gyors elérésére alkalmas.

Merevlemez tárolás = Merevlemez diszk (HARD DISK, RIGID DISK)

Olyan mágneslemez tárolás, amely mágneses réteggel borított kerámia- vagy alumíniumlemezből készül. A merevlemez tárolás nagy tömegű adat tárolására, és az adatok közvetlen, viszonylag gyors elérésére alkalmas.

Midi torony

Álló számítógépház, a mini toronynál egy 5,25"-os periféria rekesszel nagyobb.

Midrange computer

L.: Közepes számítógép

Mikroprocesszor

Olyan nagy bonyolultságú félvezető eszköz, amely a digitális számítógép központi egységének (CPU) a feladatait végzi el. A mikroprocesszor dekódolja és végrehajtja az utasításokat, vezérli a műveletek elvégzéséhez szükséges belső adatforgalmat és a csatlakozó perifériális berendezések tevékenységét.

Mikroprogram (MICROPROGRAM)

A gépi utasítás elemi lépéseinek végrehajtását vezérlő program, amelyet általában egy csak olvasható tárolásban, a mikroprogramtárolásban tárolnak. Mikroprogramozott számítógépeknél az adott utasítás kódja alapján választódik ki a mikroprogramtárból az a mikroprogram, amely a vonatkozó utasítást megvalósítja.

Mikroprogram vezérlőegység = MCU (MICROPROGRAM CONTROL UNIT)

Vertikális mikroprogramozásnál a mikroprogram lefutását vezérlő áramkörök összessége. A CISC processzorok vezérlőegységének része.

Mikroprogramtár (MICROPROGRAM STORAGE)

A mikroprogramozott műveleti vezérléshez szükséges mikroutasításokat tartalmazó tároló.

Mikroszámítógép (MICROCOMPUTER)

Egy lapkából álló mikroprocesszorra épülő számítógép. Annak ellenére, hogy kisebb teljesítményű a miniszámítógépeknél és a nagyszámítógépeknél, a mikroszámítógép bonyolult feladatok megoldására alkalmas hatékony eszközzé fejlődött.

Mikroutasítás-számláló

A vertikális mikroprogramozásnál a mikroutasítások soros végrehajtásához szükséges

regiszter, amely a következő végrehajtandó mikroutasítás címét tartalmazza.

Million Instructions Per Second

L.: MIPS

Million Operations Per Second

L.: MOPS

Millions of Floating Point Operations Per Second

L.: MFLOPS

Mintavétel (SAMPLING)

Az az eljárás, amellyel egy folytonosan változó mennyiséget csak bizonyos időpontokban (a mintavételi időpontokban) felvett értékeivel ábrázolnak.

Mip-mapping

A 3D grafikában alkalmazott eljárás, amelyet a korszerű monitorvezérlő kártyák hardverében is realizálnak. Alkalmazásánál egy textúrát több különböző felbontásban is letárolnak. Ha a nézőpont közeledik egy textúrázott objektumhoz, akkor a pixelesedés kizárása érdekében egyre finomabb felbontású változatát feszítjük fel a textúrának az objektum felszínére. (Minél közelebb van egy objektum, annál részletesebb a textúra, a távolabbi tárgyak pedig elmosódottabbak.)

Monitor (megjelenítő)= Megjelenítő(MONITOR)

Személyi számítógépekhez alkalmazható katódsugárcsöves eszköz, amely az adatok vizuális megjelenítésére szolgál.

Monitorvezérlő kártya

A monitorok meghajtásához szükséges áramkörök nagy részét tartalmazó illesztőkártya. A kártyán található többek között egy grafikus memória, amely a képernyőn megjelenő képpontok kódjait tartalmazza, és egy céláramkör, amely digitális-analóg konverzióval a monitor elektronágyúinak a vezérlését látja el.

More (DOS parancs)

A bemeneti információt oldalakra törve írja ki a képernyőre.

Move (DOS parancs)

A fájlok mozgatására szolgál. A copy-tól annyiban tér el, hogy az eredeti állományt le is törli, de az áthelyezést csak logikailag (a nyilvántartásban) végzi el.

Multimédia

Hang, grafika, animáció és mozgókép együttes alkalmazása. A weben a multimédia a hipermédia része, amely az előzőleg említett elemeket még a hiperszöveggel is kombinálja.

Multimédiás művelet = MMX utasítás

A multimédiás képfeldolgozáshoz szükséges (SIMD) programutasítás, amely MMX adattípusokat dolgoz fel.

Multiplexed transaction

L.: Multiplexált tranzakció (adatátvitel)

Multiplexált cím- és adatátvitel

L.: Multiplexált tranzakció (adatátvitel)

Multiplexált tranzakció (adatátvitel) (MULTIPLEXED TRANSACTION)

Az adatátvitelnek az a módja, amikor a címek és adatok váltakozva, egymás után ugyanazon a vezetékeken keresztül kerülnek átvitelre.

Mutató

L.: Közvetett címzés

Mágneslemezes tár= Diszk, mágneslemezes egység

(MAGNETIC DISK MEMORY/STORAGE/STORE, DISC, MAGNETIC DISK)

Olyan tár, amely mágneses lemezt használ tárolóközegként. Van merev és hajlékony lemezeket használó, beépített vagy cserélhető, egy vagy több mágneslemez tartalmazó mágneslemezes tár. Mozgófejes tárnál az adatok beírása, kiolvasása a lemezek közé benyúló mozgó mechanizmussal ellátott író-olvasó (mágnes)fejekkel történik. A mágneslemez felülete koncentrikus sávokra van felosztva, és minden lemezoldalhoz egy vagy több mágnesfej van hozzárendelve.

Mágnesszalag kazetta

L.: Kazetta

Kazetta= Mágnesszalag kazetta(CARTRIDGE, CASSETTE)

Adat vagy program tárolására alkalmas mágnesszalagot tartalmazó kazetta.

Második szintű cache = L2 (LEVEL 2 CACHE)

A processzoron kívüli (off-chip) gyorsító tároló, amely általában az alaplapon kerül elhelyezésre. (A P II, P III és egyes Celeron processzoroknál a tokba beépített).

Másodlagos gyorsító tár

L.: Második szintű cache

Mátrix (MATRIX)

1. Kétdimenziós derékszögű alakzatban rendezett adatok tömbje, mátrixalgebra szabályai szerint kezelhető. A mátrixokra utasításokkal lehet műveleteket előírni.

Mátrixnyomtató (DOT-MATRIX PRINTER)

Olyan nyomtató, amely pontokból álló karaktereket nyomtat tűs nyomtatófej segítségével.

Működési felügyelet (HARDWARE MONITORING)

Biztosítja az alaplapon működésének programmal történő automatikus és folyamatos ellenőrzését. Jelzi a felhasználó számára a hardver egyes meghibásodásait (például túlmelegedés).

Műveleti utasítások

Az utasításkészletnek azok az utasítástípusai, amelyekkel aritmetikai, logikai, bitorientált és multimédiás műveletek hajthatók végre.

Csoportjai:

- aritmetikai utasítások (ADD, SUB, MUL, DIV stb.),
- logikai utasítások (AND, OR, XOR stb.),
- bitléptető/forgató utasítások (SLL, SRL stb.),
- bitműveletek, karakterlánc (string) műveletek (CMPB stb.),
- multimédiás és 3D grafikus műveletek (PADD, PADDs stb.).

Műveleti vezérlés

A gépi utasítások elemi lépéseinek végrehajtása során a számítógép összes, a műveletvégrehajtáshoz szükséges hardverrészegységének a gépi utasítás alapján történő összehangolt irányítása.

Műveletkód (OPERATION CODE)

A számítógép elemi műveleteinek ábrázolására szolgáló kód. Az utasításnak az a része, amely megadja a gép számára, hogy milyen műveletet (pl. összeadás, osztás, összehasonlítás) kell elvégeznie.

Műveletvégző társprocesszor

Önálló működésre alkalmas processzor, amellyel az architektúrát a processzor tehermentesítése érdekében bővítjük. Utasítások, parancsok küldésével "felprogramozható", ezután teljes önállósággal vezérli az egyes részfeladatok végrehajtását. Általában célfeladatokra alkalmazzák (például lebegőpontos társprocesszor).

N-utas csoport asszociatív cache = Set associative cache (N-WAY SET ASSOCIATIVE)

Olyan cachetároló, amely több, "n" sorból álló csoportokra van osztva, és az egy csoporthoz tartozó cachetárolórész önmagában teljesen asszociatív tárolóként működik. Annak megállapítása, hogy egy cachebe írandó blokk melyik csoporthoz tartozik, a közvetlen leképezésű cachehez hasonlóan, a memóriacíméből képzett indexszel kerül meghatározásra.

N-way set associative cache

L.: n-utas csoport asszociatív cache

NETPC

Hálózati munkaállomásként működtethető, egyszerűsített mikroszámítógép. A szokásos PC funkciók jelentős részét hálózati szolgáltatásokkal helyettesítik, így a NETPC hardver-, szoftver-architektúrája jelentősen leegyszerűsíthető, és ezáltal áruk a szokásos mikroszámítógépekénél lényegesen olcsóbb lehet. Alkalmazásuk feltétele a nagy átbocsátóképességű hálózat kiépítése.

NI-CAD akkumulátor

A gomelemek használata előtt ez a lítium-kadmium akkumulátor biztosította a CMOS óra áramellátását. Élettartama a műszaki leírások szerint egy év, gyakorlati tapasztalatok alapján háromévente volt tanácsos cserélni. A számítógép használata során töltődött az akkumulátor, kikapcsolása után a CMOS óra készenléti állapotba került, amelyhez az akku biztosította a tápfeszültséget.

NLX (Now Low Extended)

Ipari szabvány a helytakarékos számítógép kisméretű alaplappjának fizikai felépítésére. Meghatározza az alaplapp, a foglalatok méretét, a kártyahelyekbe helyezett kártyák

magasságát az alaplapon.

NOP utasítás (pipeline) (NO OPERATION)

Üres utasítás, végrehajtása nem változtat a számítógép állapotán.

A pipeline szervezése során a memóriautasítások végrehajtásához szükséges többlet-időigény és a hazardok miatti utasítás-várakoztatás NOP utasítások beiktatásával oldható meg.

NOT (Logikai művelet)

Az operandus minden bitjét negálja (egyes komplement).

Az utasítás működése: A NOT utasítás az operandus minden bitjét ellenkezőjére állítja be és ezt az operandus címén helyezi el. Az utasítás végrehajtása a flag-eket nem befolyásolja.

Nagyszámítógép= Host (hálózatokban)(MAINFRAME)

A számítógépek teljesítmény szerinti osztályozásának egyik kategóriája.

Jellemzői:

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

Felhasználásukra példák: nagy szervezetek nagy megbízhatóságú rendszerei (katonaság, rendőrség, bankok); nagytömegű adatot feldolgozó adatbázisszerverek; médiaszerverek; bonyolult jelenségek szimulációja (például időjárás-előrejelzés).

Nem felejtő tároló

L.: ROM

Nem maszkolható megszakítás = NMI (NON MASKABLE INTERRUPT)

A program futását átmenetileg felfüggesztő események két kategóriája közül az egyik. Olyan esemény, amely megszakítási igénye nem tiltható le, és minden esetben ki kell szolgálni. (Például: súlyos hardverhibák.)

Nem strukturális gyorsítás

A számítógéprendszer architektúrák teljesítmény növelésének egyik alapvető módszere.

Formái:

- órajel növelése;
- a program optimalizált fordítása (ennek különösen a RISC processzoroknál van jelentősége).

Nem szám

Az IEEE 754-es lebegőpontos szabvány szerinti számábrázolási formátum. A mantissza tetszőleges nem 0 értékű szám, a karakterisztika minden bitje 1-es értékű.

Network adapter

L.: Hálózati csatoló

Neumann elvek

Az első számítógépek működésének és felépítésének Neumann János által megfogalmazott elvei:

- számítógép működését tárolt program vezérli;
- a vezérlés vezérlésáramlásos (control-flow);
- a gép belső tárolójában a program utasításai és a végrehajtásukhoz szükséges adatok

egyaránt megtalálhatók (közös utasítás és adattárolás, a program felülírhatja magát);
- az aritmetikai és logikai műveletek (programutasítások) végrehajtását önálló részegység (ALU) végzi;
- az adatok és programok beolvasására és az eredmények megjelenítésére önálló egységek (perifériák) szolgálnak.

Neurális hálózat = Sejtautomaták, mesterséges intelligenciahálózat, tudasalapu a

Az emberi ideghálózat működésének modellezésén alapuló számítógépek. Lényegük a neuronok (sejtek) közötti kapcsolat, amelyet az úgynevezett szinaptikus súlyokkal lehet vezérelni.

Az emberhez hasonlóan taníthatóak, képesek nagyfokú általánosításra (absztrakcióra), és asszociációra.

Alkalmazásuk az alakfelismerés, képfeldolgozás, hangfelismerés stb. felhasználási területeken már a gyakorlatban is megkezdődött.

Notebook computer

L.: Noteszgép

Notepad= PEN-PC, penbook

Olyan hordozható számítógép, amelyen sem billentyűzet, sem egér, sem trackball nem található. Külsőjét tekintve egy jegyzettömbhöz hasonlít leginkább. Az adatbevitel és a számítógép vezérlése egy elektronikus tollal (pen) történik. Beviteli és megjelenítő felülete maga a monitor.

Noteszgép (NOTEBOOK COMPUTER)

Kis hordozható számítógép. Alakja egy összecusokott füzethez hasonló, az egyik élén forgópántokkal. Sík LCD képernyőt és kis billentyűzetet tartalmaz.

Now Low Extended

L.: NLX

Numeric coprocessor

L.: Lebegőpontos processzor

Numerikus (NUMERICAL)

A csak számjegyekből álló karaktersorozatokat numerikusnak nevezzük.

Nyomkövetés (DEBUG)

Általában tesztelési célból egy program olyan futtatása, amely során rendszeresen felfüggesztik a program működését, és például egyes tárolótartalmak kiírását követően folytatják csak a futtatást.

Nyomtató (PRINTER)

Olyan kiviteli egység, amely nyomtatással papíron jeleníti meg az adatokat. Főbb fajtái a következők: mechanikus nyomtatók (mátrixnyomtató, margarétafejes nyomtató, sornyomtató) és nem mechanikus nyomtatók (hőnyomtató, tintasugaras nyomtató, lézernyomtató).

Nyílt architektúra

L.: Nyílt rendszer

Nyílt rendszer = Nyílt architektúra

Olyan rendszer, amelynek felépítését és működését nyilvános, szakmailag elfogadott szabványok határozzák meg.

OR (Logikai művelet)

Két operandus között logikai „VAGY” művelet végrehajtása bitről-bitre.

Az utasítás működése: Az eredmény azokban a bitpozíciókban lesz 0, ahol mindkét operandusban 0 található. Minden más esetben az eredmény 1. A művelet eredménye a céloperandus helyére kerül beírásra és ZF beállításra kerül.

Object code

L.: Tárgykód

Off-chip cache

Második vagy harmadik szintű (L2, L3) cachetároló, amely a processzoron kívül helyezkedik el.

Off-line kapcsolat

L.: Offline

Offline(OFF-LINE)

Számítástechnikai szóhasználatban az online ellentéte.

Olyan számítógépes rendszerhez tartozó berendezés, illetve az ilyen berendezés által végzett tevékenység jelzője, amelyet a számítógép nem vezérel közvetlenül, illetve amellyel nincs adatátviteli kapcsolatban.

Olvasási jog (tárolóvédelem)

Egy taszk felhatalmazása arra, hogy egy szegmens vagy lap adatait olvassa.

On chip cache

L.: Elsőszintű cache

On-line (ONLINE)

Számítástechnikai szóhasználatban az offline ellentéte.

Olyan berendezés vagy az általa végzett tevékenység jelzője, amellyel a számítógép aktív adatátviteli vagy vezérlési kapcsolatban van.

Előfeltétele a közvetlen vonali kapcsolat. Például online adatbevitel csatlakoztatott terminálokon keresztül.

On-line kapcsolat

L.: On-line

Operandus (OPERAND)

Valamilyen művelet tárgyát képező objektum, azaz adat vagy annak nyelvi reprezentánsa. Az operandusok gépi szinten adatok. pl. egy fixpontos szám, adott hosszúságú bitfűzés, nyelvekben pedig azok nyelvi reprezentánsai, amelyek lehetnek közvetlen operandusok (literálok) vagy változók.

Operandus hivatkozás

A gépi utasítás része, megadja a processzor számára, hogy hol található, illetve milyen adatokkal kell a műveletet végrehajtani.

Operation code

L.: Műveletkód

Operatív tár = Központi tár, főtár, műveleti tár, rendszermemória

(MAIN MEMORY/STORAGE/STORE, CENTRAL MEMORY/STORAGE, PRIMARE MEMORY/STORAGE/STORE, OPERATING MEMORY)

A programból közvetlenül címezhető, írható-olvasható (RAM) és csak olvasható (ROM) részekből álló tár. Tartalmazza a végrehajtás alatt álló programrészeket, az ezekhez szükséges adatokat és az operációs rendszer aktív részeit.

Operációs rendszer betöltése = Indítás, boot, kezdeti betöltés, rendszerindítás

A számítógép bekapcsolásakor végbemenő folyamat. Amikor a számítógépet bekapcsolják a BIOS beolvassa a Master Boot rekordot, és elindítja a teljes operációs rendszer betöltését és elindítását végrehajtó programot.

Operációs rendszer

A gép alpműködtetését a felhasználás céljától függetlenül biztosító, általános (például egy eszköz hibás működésének behatárolását, erről a felhasználó tájékoztatását is elvégző) programok együttese.

Optikai lemez = CD, kompakt diszk(COMPACT CISC)

Optikai jelrögzítésen alapuló adathordozó.

8 vagy 12 cm átmérőjű műanyag korong, amelyen 650-1300 MB (DVD technikával 17 GB) adat tárolható. Készülhet csak olvasható (CD-ROM) illetve egyszer vagy többször írható változatban.

P3

L.: Pentium III mikroprocesszor

P6 architektúra

Az Intel PentiumPro, Pentium II és III Risc maggal rendelkező processzorának közös architektúrája.

P6 processzorcsalád

A P6 architektúrájú maggal rendelkező processzorok együttes megnevezése.

PC (Personal Computer)

L.: Személyi számítógép

PC100 SDRAM (PC100 Synchrones Dynamic Random Access Memory)

100 MHz frekvencián működő rendszerbusz órajeléhez szinkronizált dinamikus RAM típus.

PC133 SDRAM (PC133 Synchrones Dynamic Random Access

Memory)

133 MHz frekvencián működő rendszerbusz órajeléhez szinkronizált dinamikus RAM típus.

PCI 2.0 (Peripheral Component Interconnect Bus 2.1 version)

A PCI busz 2.0 verziójának specifikációja 1993-ben jelent meg.

PCI 2.1 (Peripheral Component Interconnect Bus 2.1 version)

A PCI busz 2.1 verziójának specifikációja 1995-ben jelent meg.

PCI 2.2 (Peripheral Component Interconnect Bus 2.2 version)

A PCI busz 2.2 verziójának specifikációja 1999-ben jelent meg.

PCI Bridge

A Front Side Bus-t és a PCI sít összekapcsoló kommunikációs hardveregység, amely az alaplap áramkörkészlet (chipset) része. Lehetővé teszi a PCI sínrendszer "processzorfüggetlen" alkalmazását, és két PCI egységnek a PCI Bridge-en keresztüli adatcseréjét.

PCI SIG (Special Interest Group)

A PCI sínszabványt gondozó szakmai szervezet megnevezésének rövidítése.

PCI busz (Peripheral Component Interconnect Bus)

Az Intel cég által kifejlesztett 32 bites cím, 32 vagy 64 bites adat egyidejű átvitelére alkalmas I/O buszrendszer. A ma használatos PC-kben elterjedten alkalmazott ipari sínrendszer szabvány, amely szerint a processzor és a PCI busz között elhelyezett bridge lehetővé teszi, hogy a buszrendszer a konkrét processzortól és annak sebességétől függetlenül is működőképes legyen.

PCI busz kártyahelye

A ma használatos PC-kben elterjedten alkalmazott PCI sínrendszer szabvány szerint készített, az alaplapon található csatlakozóhely, amely vezérlő és adapter áramköri kártyák csatlakoztatására szolgál.

PCI hostbridge

L.: PCI Bridge

PCI sín slot-ja

L.: PCI busz kártyahelye

PCI sín

L.: PCI busz

PCI-ISA bridge (Peripheral Component Interconnect Bus - Industry Standard Archi

Az ISA és a PCI sínrendszert összekötő bridge áramkörök összessége.

PCI-X

A PCI-X szabvány 1.0-ás verzióját a legnagyobb szervergyártók kezdeményezésére, a szélessávú összeköttetést igénylő interfészek (Gigabit Ethernet, üvegszál-összeköttetés, Ultra3SCSI) miatt 1999-ben dolgozták ki. Ennek továbbfejlesztett 2.0-s verzióját 2002 júniusában publikálták. Maximális buszfrekvencia 133 MHz; adatátvitel ciklusonként 64 bit; Maximális adatátviteli sebesség 1048 Mbájt/sec.

PCI

L.: PCI busz

PIO

L.: Programozott bevitel/kivitel

POST (Power-On Self Test)

L.: Bekapcsolási önteszt program

PS/2

Billentyűzetek és egerek csatlakoztatására szolgáló port illetve csatlakozó típus.

Page Frame

L.: Lapkeret

Page

L.: Lap

PageUp/PageDown (billentyű)

Ahol használható, ott lapozni lehet a képernyőn felfelé illetve lefelé egy oldalt.

Paging

L.: Lapozás

Palmtop

L.: Zsebszámítógép

Parallel port

L.: Párhuzamos port

Parancsregiszter (COMMAND REGISTER)

Az I/O eszközevezérlőben található átmeneti tároló, amelybe a processzor (programutasítás) az I/O művelethez szükséges vezérlési (parancs) adatokat beírja.

Parancsértelmező (COMMAND INTERPRETER)

Az operációs rendszernek parancsokat fogadó, és azokat végrehajtó része. A parancsértelmező végzi az alkalmazások betöltését, és biztosítja az alkalmazások közötti adatcserét. Emellett esetenként egyszerűbb rendszerfeladatokat is képes ellátni, például állományokat másolni és mozgatni, könyvtárlistákat kiírni.

Parity bit

L.: Paritásbit

Paritásbit (PARITY BIT)

Számítógéprendszerekben belül vagy azok között átvitt adatbitekből álló csoportok hibaelőírására használt külön bit.

Pascal (programnyelv)(PASCAL)

Magas szintű általános célú programozási nyelv, amely hatékonyan támogatja a strukturált programozást. Egyszerű megvalósíthatósága miatt széles körben elterjedt a mini- és mikroszámítógépeken. Van konkurens programozást támogató változata is. Az ALGOL-60 továbbfejlesztésének tekinthető. Kifejlesztése a 70-es évek elején, N.Wirth nevéhez fűződik.

Path (DOS parancs)

Az elérési út megtekintése, megváltoztatása. Az AUTOEXEC.BAT-ban használják.

Pause (DOS parancs)

Batch fájlokban vár egy billentyű leütésére.

Pause/Break (billentyű)

A Pause gomb megnyomására a program futása leáll és csak egy billentyű leütésére folytatódik. A Ctrl+Break (vagy Ctrl+C) megállítja a futó programot és a vezérlés visszakerül az operációs rendszerhez. Egyes programok ezeket a funkciókat le tudják tiltani.

Pentium 4 mikroprocesszor = P4

Az Intel 2000 novemberében kihozott P7 architektúrájú processzora.

Fontosabb jellemzői:

- RISC mikROUTASÍTÁSOKAT TARTALMAZÓ L1 szintű utasításcache;
- Hiper pipeline: 20 lépéses futószalag;
- SSE2 utasításkészlet.

Pentium II mikroprocesszor = P2

A piacon 1997 májusában megjelent, MMX technológiát alkalmazó P6 architektúrájú processzor.

A CPU mag 7,5 millió tranzisztort tartalmaz.

Fajtái: Celeron, Deschutes, Xeon, Katmai.

Pentium III mikroprocesszor = P3

Az Intel 1999-ben kihozott P6 architektúrájú processzora. Az Intel célja az utasításkészlet bővítése volt a 3D grafika igényei szerint. Az SSE (Streaming SIMD Extension) lebegőpontos utasításkészlet-bővítés új utasításai:

- 50 új SIMD utasítás a 3D grafikához;
- 12 új multimédiás utasítás;
- 8 új cachekezelő utasítás.

Az új utasításokat 8 db új 128 bites lebegőpontos regiszter támogatja.

Pentium MMX mikroprocesszor

Az Intel utolsó ötödik generációs processzora a Pentium MMX.

A legnagyobb eltérés a Pentium és a Pentium MMX között az MMX utasításkészlet-bővítés. Az olyan alkalmazások futtatása során, amelyek az új MMX utasításkészletet

használják, nagy sebességkülönbséget tapasztalhatunk.

Pentium MMX processzor

L.: Pentium MMX mikroprocesszor

Pentium gépcsald

Az Intel PentiumPro, Pentium MMX, Pentium II , III vagy 4 processzorára épülő számítógépek közös megnevezése.

Pentium mikroprocesszor

Az Intel cég első superskalár (azaz egy órajelciklus alatt több utasítás végrehajtására is képes) processzora, amely 1993-ban jelent meg. Első változatai 3,1 millió tranzisztort tartalmaztak, 60 MHz-es órajellel működtek. Adatbusza 64, címbusza 32 bites, 2 db 8 Kbájtos L1 cachetárolót tartalmaz.

Periféria = Periférikus egység (PERIPHERAL, PERIPHERAL UNIT, PERIPHERAL EQUIPMEN

A számítógépnek adatbevitelre illetve adatkivitelre szolgáló egysége.

Perifériainterfész

L.: I/O interfész

Peripheral Component Interconnenct local bus

L.: PCI busz

Personal Computer

L.: Személyi számítógép

Pheripheral Component Interconnect bus

L.: PCI busz

Pipelined burst üzemmód

Többszavas adatátviteli forma, amikor egy címfázist több adatfázis követ. Az első szó átvitelét követően a második, harmadik stb. szó címét automatikusan generálja íráskor az adatfogadó, olvasáskor az adatküldő eszköz.

A Pentium gépcsaldba tartozó processzorok multiplexált cím- és adatátvitelt megvalósító rendszerbuszának gyorsított átviteli módban (Burst-Modus) való működése. Ennek során a busz egy címküldést követően egy cacheadatsor méretű blokkot másol be az L1 cachebe, vagy ír vissza a memóriába vagy L2 cachebe.

Pipelining = Csővonal feldolgozás, adatcsatornás feldolgozás, futószalag feldolg

Az utasításszinten párhuzamos működésű processzorok átlapolt utasítás-feldolgozását lehetővé tevő rendszertechnikai eljárás. Alapgondolata: mivel a gépi utasítások elemi lépései (előkészítés, dekódolás, operandusok címszámítása stb.) különböző hardvererőforrásokat igényelnek, ha egy utasítás egy elemi lépése végrehajtásra került, akkor az ehhez szükséges hardveregység felszabadult, és így ez igénybe vehető egy következő utasítás elemi lépésének végrehajtására.

Pixel (Picture element)

Képpont, a raszteres kép elemi egysége.

Platformfüggetlen

L.: Hordozható szoftver

Plug and Play = PnP, csatlakoztasd és használd

Magyarra fordítva: csatlakoztasd és használd.

A számítógéprendszer PnP képessége azt jelenti, hogy a működés közben csatlakoztatott I/O eszközöket a rendszer képes automatikusan felismerni, és ezt követően kezelni.

Az operációs rendszer és a PnP támogatására felkészített hardverelem egy közös nyelven kommunikál, az eszköz az operációs rendszer kérdéseire megfelelő válaszokat ad, azonosítja magát, és megadja a számára szükséges kiszolgáló rutinok adatait, illetve az operációs rendszer automatikusan kijelöli számára a megszakítást, a címtartományt, és telepíti a megfelelő fájlokat.

Plug and Play Association

Ez a szervezet gondozza a Plug and Play ipari szabványt. Az egyesületben található cégek 3Com, Compaq, IBM, Intel, Microsoft.

PnP

L.: Plug and Play

Pointer

L.: Közvetett címzés

Polling (programozott I/O) = PIO

Az az eljárás, amikor az operációs rendszer periodikusan és ismétlődően lekérdezi az I/O egység állapotregiszterének bitjeit annak érdekében, hogy az eszköz I/O műveletre való alkalmasságát vagy az adatátvitel hibátlan lefutását megállapítsa.

Port

A számítógépnek az a része, amely a perifériális eszközökkel tart kapcsolatot. A portot a processzor egy vagy több címezhető tárolóhelynek látja, amelyeket az I/O eszközökkel való kommunikációban adatok küldésére vagy fogadására tud használni.

Pozicionálási idő (mágneselem)

Az az idő, amely ahhoz szükséges, hogy a mágneslemez író/olvasó feje ráálljon a megcímezett sávra.

Preemption

Dinamikus buszhasználat esetén a lefoglalt busz felszabadításának egyik eljárása.

Alkalmazása esetén ha egy magasabb prioritású master buszfoglalási igényt jelez, akkor ez a folyó busztranzakciót megszakítja.

Prefetching

Az egyik gyakori cachebe való blokkbemásolási eljárás. Jellemzője, hogy ha a főtárból be kell tölteni egy blokkot a cachebe, akkor automatikusan betöltésre kerül a főtár következő blokkja is. (Feltételezhető, hogy ha a processzornak szüksége volt egy blokk adataira, akkor nagy valószínűséggel szükség lesz a rákövetkező blokk adataira is.)

Prioritásos multiprogramozás

A multiprogramozott üzemmód egyik változata, amelynél a programfolyamatoknak prioritása van, és ennek megfelelő sorrendben kapják meg a számítógép erőforrásait (processzor, memória, I/O vezérlések).

Privilegizált utasítás = Kitüntetett utasítás (PRIVILEGED INSTRUCTION)

A processzor utasításkészletének azok a speciális utasításai, amelyeket a processzor csak az operációs rendszer privilegizálási szintjén futó taszkok esetén hajt végre.

Privilegizálási szint (tárolóvédelem)

Hierarchikus tárolóvédelmi rendszerben a taszkok jogosultságait meghatározó besorolás, amely minden egyes taszkhoz hozzárendelésre kerül.

Napjaink számítógépes rendszereiben minimálisan két privilegizálási szintet különböztetnek meg:

- egy magas jogosultsági szintet az operációs rendszerhez tartozó programok számára;
- egy alacsonyabb jogosultsági szintet a felhasználói programok számára.

Probléma-orientált nyelvek

Egy adott feladatcsoport, felhasználási terület logikáját a lehető legjobban megvalósító, így a programozást megkönnyítő programnyelv.

Process control mode

L.: Folyamatvezérelt üzemmód

Processor-PCI Bridge

A PCI sínrendszer ipari szabványnál a processzor és a PCI busz közötti kapcsolatot biztosító áramkörök együttese. Lehetővé teszi, hogy a PCI buszrendszer a konkrét processzortól és annak sebességétől függetlenül is működőképes legyen.

Processzor (PROCESSOR)

A számítógép műveletvégző egysége, amely utasításokat értelmez és hajt végre.

Processzor-konzisztencia (szuperskalár processzorok)

A soros konzisztencia egyik összetevője, amely a párhuzamos utasítás-végrehajtásnál a programutasítások eredeti sorrendben történő végrehajtását jelenti.

Processzorfoglalat

Alaplapon található részegység, amely a processzor csatlakoztatására szolgál. Minden processzorcsaládnak más és más foglalatra van szüksége, ezek fizikai mérete és a lábak száma eltérő. Az alaplapi processzorcsatlakozónak két alaptípusa alakult ki, az egyik foglalat a socket, míg a másik a slot nevet kapta.

Program (PROGRAM)

A számítógép számára értelmezhető és végrehajtható utasítások sorozata.

Program compability

L.: Program-kompatibilitás

Program counter

L.: Utasítás-számláló regiszter

Program-kompatibilitás (PROGRAM COMPATIBILITY)

Két különböző számítógép akkor programkompatibilis, ha az egyikre írt programok a másikon minden változtatás nélkül futtathatóak.

Programfolyamat

L.: Taszk

Programlokalitás elve

A programok egy kis időintervallumban a címtérnek, a memóriának nagy valószínűséggel csak egy relative kis részét veszik igénybe.

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

Helyi lokalitás: Ha egy adatra vagy egy utasításra hivatkozás történik, akkor nagy valószínűséggel ez a környezetében lévő címekre is megtörténik (soros utasítás-végrehajtás).

Programmegszakítás

A programvégrehajtás időszakos felfüggesztése.

A ma használatos processzoroknál a felfüggesztő okok két osztálya a megszakítások (interrupt) és a kivétel (exception).

Programozott I/O

L.: Programozott bevitel/kivitel

Programozott bemeneti/kimeneti (I/O) adatátvitel

Az összes adatátvitellel kapcsolatos műveletet a számítógép programja vezérli.

Programozott bevitel/kivitel = Programozott I/O, PIO (PROGRAMMED INPUT-OUTPUT))

Az I/O műveletek végrehajtásának egyik módszere, amikor az adatátvitelt az I/O eszköz és a főtár között egy program vezérli. A processzor az I/O művelet teljes ideje alatt foglalt.

Programozott kivétel

A kivételek egyik típusa, amelyet az Intel és vele kompatibilis processzorok utasításkészletében meglevő INT és IRET utasítások eredményeznek.

Programvezérelt megszakítás

L.: Programozott bevitel/kivitel

Proxy szerver (LAN/WAN)

A világháló, és a helyi hálózatok közötti gyorsítótár. Segítségével a gyakran letöltött fájlok, web oldalak a lokális hálózat sebességén érhetők el.

Pufferelt közvetlen átírás (BUFFERED WRITE THROUGH)

A cacheben megváltoztatott adatoknak főtárba való visszaírására alkalmazott közvetlen

átírásnak a hatékonyságát javító módszer.

A processzor a cacheben megváltozott főtárbeli adatokat egy íráspufferbe írja be, és nem várja meg a főtár írásának a befejeződését.

Pufferregiszter (BUFFER REGISTER)

Az adatok áramlási ütemének egyenetlenségeit, vagy az események időbeli előfordulásának különbségét egyenlíti ki abban az esetben, amikor az adatokat különböző sebességű hardveregységek között kell mozgatni.

Párhuzamos adatátvitel

Az adatátvitelnek az a típusa, amikor a periféria interfész és a periféria között az adatokat bitcsoportonként egyszerre visszük át.

Párhuzamos buszfoglalási eljárás

A központosított buszarbitráció egyik módszere, amikor minden sínhasználatért folyamodó master eszköz önálló kérő ("request") és engedélyező ("grant") vezérlővonallal rendelkezik. A sínvezérlés prioritás szerint engedélyezi a sín igénybevételét, amelynek meghatározása különböző eljárások szerint történhet, például körbenjáró vagy LRU eljárással.

Párhuzamos port = Nyomtató port, LPT LINE PRINTER TERMINAL, PARALLEL PORT, PRINT

A perifériális egységek csatlakoztatására szolgáló interfészek egyik típusa. A párhuzamos port az átküldendő bájt bitjeit egyszerre, több adatvezetéken egy időben továbbítja. A párhuzamos portra legtöbbször a nyomtatót csatlakoztatjuk.

Párhuzamos port= Nyomtató port, LPT

LINE PRINTER TERMINAL, PARALLEL PORT, PRINTER PORT, CENTRONICS PORT, LPT PORT

A perifériális egységek csatlakoztatására szolgáló interfészek egyik típusa. A párhuzamos port az átküldendő bájt bitjeit egyszerre, több adatvezetéken egy időben továbbítja. A párhuzamos portra legtöbbször a nyomtatót csatlakoztatjuk.

Párhuzamos utasítás dekódolás (szuperskalár processzorok)

A párhuzamosan működő végrehajtó egységek megfelelő számú utasítással való ellátása érdekében a szuperskalár működésnél egy időben több programutasítást is dekódolni kell.

RAID (Redundant Array of Independent Discs)

Több merevlemezről álló lemeztömb, amelyen az adatok redundáns elhelyezése csökkenti a hozzáférési időt, és megnöveli a működés biztonságát (bizonyos mennyiségű adat megsérülése esetén a redundáns információ alapján az elveszett adatok visszanyerhetők).

RAM (Random Access Memory)

Közvetlen elérésű írható-olvasható tár, tartalmát a számítógép kikapcsolásakor elveszti.

RAS (Row Address Strobe) = Sorcím kiválasztó impulzus

A multiplexált memóriacímzéshez szükséges vezérlőjel, amely jelzi a DRAM csipnek, hogy a megfelelő sorcím a címpufferbe elhelyezésre került.

RBS (Return Stack Buffer)

A szubrutinból való visszatérési címeket kezelő gyorsítótár, amelyet a spekulatív elágazás-feldolgozáshoz használ a processzor.

REQ (Request)

L.: Buszfoglalási kérelem

RISC (Reduced Instruction Set Computer)

L.: Csökkentett utasításkészletű számítógép

ROB (Re Order Buffer)

L.: Átrendező puffertár

ROM (Read Only Memory)= Csak olvasható tár

Olyan tár, amelynek tartalma üzemszerű használatkor csak olvasható, tartalma a számítógép kikapcsolásakor is megőrződik. Leggyakrabban mikroprogramtárként és a BIOS tárolójaként alkalmazzák, tartalmát általában a gyártó írja be.

ROM-BIOS (Read-Only Memory Basic Input/Output System)

Csak olvasható memória, amely a huzamosabb időn keresztül megőrzendő alapvető eszközkezelő rendszerprogramokat tárolja.

RS-232-C port

Szabvány a soros porton keresztül történő soros adatátvitelre. Részletesen meghatározza az adatátvitel mechanikai, villamos, funkcionális és eljárási szabályrendszerét.

RSB (Return Stack Buffer) = Visszatérési verem tároló

A Pentium processzorcsalád korszerűbb tagjainak a dinamikus elágazás előrejelzésre használt, a processzorba beépített egyik cächetárolója, amely a szubrutinból való visszatérési címeket tartalmazza.

RTC (Real Time Clock)

L.: Valós idejű óra

Raster Image Processor = RIP

L.: Grafikus processzor

Raszteres kép

L.: Rasztergrafika

Rasztergrafika (RASTER GRAPHICS)

Raszteres, azaz képpontokból (pixelekből) álló képet generáló és feldolgozó számítógépes eljárások összessége.

Rd (DOS parancs)

Könyvtár megszüntetése (ha üres és nem tartózkodunk benne). Lásd még deltree.

Re Order Buffer

L.: Átrendező puffertár

Register banking

L.: Regiszterbank (regisztertárak)

Register blocking

L.: Blokktechnika (regisztertárak)

Register windowing

L.: Ablaktechnika (regisztertárak)

Regiszter (processzor)

A regiszterek a processzoron belüli adatok illetve utasítások átmeneti tárolására szolgáló, nagyon gyors és drága tárolók. Méretük napjainkban általában 32, 64, 80 vagy 128 bit.

Regiszter indirekt címezése

Az utasításban szereplő regiszter tartalmazza azt a 16-bites memóriacímet, ahol az adat megtalálható.

Regiszterbank (regisztertárak) (REGISTERBANKING)

Regisztertár típus, amelyben a regisztertömb fix méretű (azaz meghatározott kettő hatványának megfelelő darabszámú regiszterből álló), át nem lapolható részekre, bankokra van felosztva, amelyeket a felhasználói taszkokhoz rendelnek hozzá.

Regiszterkészlet

A processzorban található regiszterek összessége.

Regiszterkészlet

A processzorban található regiszterek összessége.

Regisztertár

Felhasználói programok (taszkok) által használható általános regiszterek összessége. Típusai: regiszterbank, ablaktechnika, blokktechnika.

Regiszterátnevezés (szuperskalár processzorok)

Szuperskalár architektúrájú processzoroknál a programutasításokban hivatkozott regiszterek közötti áladatfüggőségek megszüntetésének egyik módja. Alkalmazása esetén az áladatfüggőséget okozó regisztert a fizikai regisztertár egy másik, nem használt regiszterével helyettesítik. Ez különösen akkor célszerű, ha a processzorba fizikailag beépített regiszterkészlet jóval nagyobb a programutasításokban használható regiszterkészletnél.

Az átnevezés folyamatának lépései:

- az utasítás kibocsátáskor a cél- és forrásregiszter átnevezése az átnevező puffertárral;
- a kiküldéskor a regiszter érvényességi bitek, azaz a függőségek ellenőrzése;
- a végrehajtás után az átnevező puffertár aktualizálása.

Rekonfigurációs képesség

Egy számítógéprendszernek az a képessége, amellyel a meghibásodott

hardveregységeket automatikusan, a folyamatos működés leállítása nélkül felismeri és izolálja, és ezek nélkül is képes tovább működni.

Relatív címzés

A gépi utasítás-címrésze az adatnak valamilyen alaplímhez vagy báziscímhez viszonyított relatív címét tartalmazza. A tényleges fizikai memóriacímet ekkor a relatív cím és a báziscím összeadásával kapjuk meg.

Release on request

Dinamikus buszhasználat esetén a lefoglalt busz felszabadításának egyik módszere. A master annyi időre foglalja le a buszt (akkor is, ha adatot nem forgalmaz), amíg a buszt másik master nem igényli.

Release when done

Dinamikus buszhasználat esetén a lefoglalt busz felszabadításának egyik módszere. A master egy tranzakcióra kapja meg a buszhasználat jogát, annak befejezése után felszabadítja a buszt.

Rem (DOS parancs)

A batch fájlokban megjegyzések (remark) elhelyezésére szolgáló parancs.

Remote bach

A köteget feldolgozásnak az a speciális esete, amikor az adatkommunikációt távadat-feldolgozással valósítják meg.

Ren (DOS parancs)

Egy fájl átnevezése (könyvtár átnevezése: move).

Rendszer (SYSTEM)

A rendszer részeit alkotó elemek és ezek kapcsolatainak olyan együttese, amelyek meghatározott ismérvek (vizsgálati célok) szempontjából összetartoznak.

Rendszerbusz = Rendszersín (SYSTEM BUS)

A processzort, a második szintű cachet és a rendszermemóriát összekötő busz.

Rendszerelem

A rendszernek azon alkotói, amelyeket a rendszer vizsgálata során már további részekre nem bontunk.

Rendszerindítás

L.: Operációs rendszer betöltése

Rendszermag= Supervisor, kernel

Az operációs rendszer központi része, amelynek feladata az erőforrások (processzor, memória stb.) elosztása és kezelése, a felhasználói folyamatok védelme és adminisztrálása.

A hardver és a felhasználói folyamatok között helyezkedik el, és a rendszermag önmaga is folyamatok sokasága. Ezek az ún. rendszerfolyamatok, amelyek a rendszer indításakor jönnek létre, és futásuk a rendszer leállításáig tart.

Rendszermemória

L.: Operatív tár

Rendszermenedzselő üzemmód = SMM (SYSTEM MANAGEMENT MODE)

A Pentium processzorcsalád energiatakarékos üzemmódja. Ebben az üzemmódban a processzor az operációs rendszertől és az alkalmazásoktól függetlenül működik.

Rendszerobjektum (tárolóvédelem)

A számítógéprendszerben egy elkülöníthető egységként kezelt program- és/vagy adatállomány.

Védett üzemmódban a rendszerobjektumok a következők:

- program- és adatszegmensek
- lapok
- taszkok
- táblázatok (LDT, TSS stb.)
- kapuk.

Rendszerregiszter

A processzor regiszterkészletének a felhasználói programok által nem "látható", nem elérhető része. Például IR utasításregiszter.

Rendszersín

L.: Rendszerbusz

Request

L.: Buszfoglalási kérelem

Reset gomb (RESET BUTTON)

A számítógépet az áram kikapcsolása nélkül újraindító kezelőszerv

Return Stack Buffer

L.: RBS

Row Address Strobe

L.: RAS

S2MP (Symmetric 2 Multiprocessors System)

Továbbfejlesztett SMP architektúra a multiprocesszoros számítógéprendszerekhez. A Silicon Graphics és a szuperszámítógépekre specializálódott leányvállalata, a Cray Research jelentette be ezt a multiprocesszoros architektúrát, amely jelentősen kibővíti a szimmetrikus multiprocesszoros (SMP) rendszerek skálázhatóságának eddigi határait.

SCSI (Small Computer System Interface) = Késszámítógépes rendszer-csatoló

Speciális, nagy teljesítményű periféria interfész, és egyben szabvány (háttértárolók, szkennerek, CD-ROM meghajtók stb.). A SCSI-sínre csatlakoztatott eszközöknek saját címük van.

Ahhoz, hogy használni tudjunk egy vagy több SCSI eszközt, szükségünk van egy kártyára, amely a PCI sínbe illesztve képes az eszközök kommunikációját vezérelni.

Ma már a klasszikus SCSI interfésznél gyorsabb, az Ultra SCSI, Wide SCSI, SCSI2 és SCSI3 névvel jelölt továbbfejlesztett változatait használják.

SDRAM (Synchrones Dynamic Random Access Memory)

Szinkronizált dinamikus RAM. Nagy hozzáférési sebességét a processzor és a memória működésének időbeli összehangolásával éri el (a RAM a memória írási és olvasási műveleteit a processzor működését meghatározó órajelhez szinkronizálja).

SIMD architektúrájú számítógép (SINGLE INSTRUCTION STREAM MULTIPLE DATA STREAM)

Olyan számítógéprendszer típus, amely egyetlen utasításfolyammal többszörös adatfolyamot dolgoz fel. Ebbe a kategóriába olyan számítógépek tartoznak, amelyek több párhuzamos működésre képes műveletvégző egységet tartalmaznak, és így vektorműveleteket képesek végrehajtani gépi utasításszinten. Például Pentium III processzor 3D grafikus utasításai.

SIR (Standard Infared)

L.: Infravörös port

SISD architektúrájú számítógép (SINGLE INSTRUCTION STREAM SINGLE DATA STREAM)

Olyan számítógéprendszer, amely egyetlen utasításfolyammal egyetlen adatfolyamot dolgoz fel.
Például Neumann elvű gépek, PC-k.

SMM (System Management Mode)

L.: Rendszermenedzselő üzemmód

SMP architektúrájú multiprocesszoros rendszer

L.: Szimmetrikus multiprocesszoros felépítés

SPEC Benchmark

Alkalmazási céltól függő tesztprogramkészlet, amely felhasználási területenként (például játékprogram, grafika, kvantummechanika) állít elő a számítógéprendszer teljesítményét jellemző mutatót.

SRAM = Statikus RAM (STATIC RANDOM ACCESS MEMORY)

A félvezető memória (RAM) egy fajtája, működési elve a billenőkör (flip-flop) logikáján alapszik. Addig megőrzi az információt, amíg az eszköz működtetéséhez az áramellátás biztosított. Az SRAM-ot általában a gyorsítótárakban használják

SSE (Streaming SIMD Extension)

Az Intel által kifejlesztett, 1999 márciusában bevezetett lebegőpontos SIMD utasításkészlet-bővítés a 3D vektorgrafikus feldolgozások hatékonyabb végrehajtásához.

Scan code

L.: Billentyűkód

Scandisk (DOS parancs)

Karbantartó parancs a háttértár hibáinak felfedezésére, megjelölésére és esetleg

kijavítására.

Scoreboarding (pipelining)

A pipelining során fellépő regiszterütközések kezelésének egyik hardveres módszere. Minden regiszterre bitek beállításával könyvelésre kerül, hogy a pipelineban van-e olyan utasítás, amely egy adott regiszterre hivatkozik. Ha egy további utasítás egy ilyen regiszterhez akar hozzáférni, akkor az késleltetésre (blokkolás, várakoztatás) kerül.

ScrollLock (billentyű)

Ezt a nagyon fontos billentyűt a legritkább esetben használják, néhány régi játékprogramban a zene ki- és bekapcsolására szolgált. Egyes táblázatkezelőknél van funkciója.

Segment

L.: Szegmens

Segédprogram (UTILITY /PROGRAM/, SUPPORT PROGRAM)

Olyan szolgáltató rutin (szolgáltató program), amely a számítógépi feldolgozás folyamatait általánosan segíti, vagyis a számítógépi program ellenőrzéséhez, végrehajtásához ad segítséget. Ilyenek a: beviteli rutinok, diagnosztikusprogramok, nyomkövető programok, rendezőprogramok, naplózóprogramok, perifériakonvertáló programok, rendszerteljesítmény-mérő programok stb.

Serial communication

L.: Soros adatátvitel

Set (DOS parancs)

A DOS belső változóinak (path, prompt, temp stb.) kiírása.

Set associative cache

L.: n-utas csoport asszociatív cache

Setup program

Számítógép operációs rendszerének (BIOS) beállításait módosítani képes konfiguráló program, ill. új szoftver telepítését indító program megnevezése.

Shift (billentyű)

Váltóbillentyű, lenyomásakor az írógéphez hasonlóan betűkkel történő használatkor NAGYBETŰK jelennek meg, számok esetén a föléjük írt jelek (!, #, % stb.).

Slave

L.: Szolga

Slot

L.: Kártyahely

Soft off

Az ATX-es számítógép olyan kikapcsolt állapota, amikor az alaplaphoz csak néhány

áramköre marad aktív állapotban. E helyzetben a bekapcsoló gomb mellett a modem, a hálózati kártya, a billentyűzet és az egér is képes újra elindítani a PC működését.

Soros adatátvitel (SERIAL COMMUNICATION)

Egy csatornán folyó, egyszerre 1 bites, azaz bitsorozatokkal megvalósított adatátvitel számítógépek vagy számítógép és perifériás eszköz között.

Soros buszfoglalási eljárás (DAISY-CHAIN)

A központosított buszarbitráció egyik módszere.

Jellemzője, hogy a mestereknek egy közös "request" és egy felfűzött (daisy-chain) "grant" vezérlővonala van.

Az eszközök sorba vannak kötve, és sorrendjük határozza meg, hogy mikor kaphatnak jogosultságot a sín használatára. A sínvezérlőhöz közelebbi master prioritása magasabb.

Soros hozzáférésű tár = Szekvenciális hozzáférésű tár (SEQUENTIAL ACCESS)

Olyan tároló, amelyből egy adatot csak a tárolóban lévő adatok egyenkénti végigolvasásával lehet kikeresni.

Szekvenciális hozzáférésű tár olvasását csak az adatállomány elején lehet elkezdeni, és addig kell folytatni, amíg a keresett adat sorra nem kerül. A szekvenciális adattárolás előfeltétele az adatok sorba rendezhetősége egy egyértelmű azonosító szerint.

Soros konzisztencia (szuperskalár processzorok)

Párhuzamos utasítás-végrehajtásnál a processzor konzisztencia és memória-konzisztencia együttese, biztosításának legfontosabb eszköze az átrendező puffertár (ROB).

Soros port = Soros interfész, aszinkron port, Serial port, COM port

A perifériális egységek csatlakoztatására szolgáló interfészek egyik típusa. Legfontosabb tulajdonsága a soros adatátvitel, azaz egy bájt bitjei időben egymás után haladnak át az adatvezetéken. A soros portra legtöbbször az egeret csatlakoztatjuk, de modem is rákapcsolható, illetve két számítógépet is összeköthetünk segítségével.

A soros interfész szabványa a V24 vagy a RS-232-C.

Soros tároló

Olyan tároló, melynél a tárolóhelyek időben egymást követően válnak hozzáférhetővé az információ beírásakor és kivitelekor.

Sort (DOS parancs)

A bemeneten megadott információt bizonyos szempontok szerint rendezi a kimenetre (pl. ABC-be).

Sound Blaster kompatibilis hangkártya

A Creative Labs által gyártott Sound Blaster (az első jó minőségű, márkás) hangkártyával kompatibilis hangkártya, amely a PC-s világban elterjedt összes operációs rendszer alatt működőképes.

Source code

L.: Forráskód

Spekulatív elágazás-feldolgozás

Eljárás, amelynek segítségével a processzor a pipelining zavartalan működése érdekében megkísérli az ugró utasítások várható irányát, kimenetét megjósolni.

Az elágazásbecslés módszere lehet:

- rögzített (egykimenetelű): a becslés mindig azonos (elágazik vagy nem)
- valódi (kétkimenetelű): statikus esetben a fordítóprogram a tárgykód elemzése alapján, vagy dinamikus esetben a program futása során a processzor az egyes elágazások megvalósulásának statisztikája alapján "jósolja meg" az elágazást.

Stand by

Az energiatakarékos működésű PC készenléti állapota, amelyben a PC nagy energiafogyasztó részeit kikapcsolásra kerülnek anélkül, 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).

Static RAM

L.: SRAM

Statikus RAM

L.: SRAM

Strukturális gyorsítás

A számítógéprendszer teljesítményének növelése az architektúra, az adat és utasításfolyamatok hatékonyabb megszervezésével. Egyik alapvető módszere a párhuzamosítás.

Párhuzamosítás a CPU-n belül:

- vektorszámítógépek;
- pipeline feldolgozás;
- szuperskalár processzorok.

Párhuzamosítás a CPU-n kívül:

- társprocesszorok;
- multiprocesszoros architektúrák (több "hagyományos" CPU-val és ALU-val).

Nem hagyományos elven működő számítógépek.

- neuronális hálók.

Subst (DOS parancs)

Egy könyvtárhoz meghajtó betűjelet rendelhetünk ezzel a paranccsal.

Synchronization

L.: Szinkronizálás (számítógép-részek)

System Management Mode

L.: Rendszermenedzselő üzemmód

System bus

L.: Rendszerbusz

System

L.: Rendszer

Szegmens (SEGMENT)

Ha a virtuális tár olyan logikai blokkokból áll, amelyeknek mérete nem rögzített, akkor

ezeket a blokkokat szegmenseknek nevezzük. A szegmensek átlapolódóan is megadhatók, azaz ugyanaz az adat két különböző szegmensen belül is megcímezhető.

Szegmenstáblázat (SEGMENT TABLE)

A szegmenseket azonosító sorszámot és a betöltött szegmensek főtárbeli fizikai kezdőcímét tartalmazó memóriatáblázat.

Szegmentált lapozásos tárkezelés

Olyan virtuális tárkezelés, amely a szegmentáláson belül alkalmazza a lapozást, azaz a virtuális tár szegmensei lapokból épülnek fel. Ezt használják védett üzemmódban az Intel és vele kompatibilis processzorok, amelyek szegmenscímezésen belüli kétszintű lapcímezést valósítanak meg.

Szegmentált virtuális tárkezelés

A virtuális tárkezelésnek az a formája, amikor a virtuális tár változó méretű blokkokból, azaz szegmensekből áll. Ennek a tárkezelésnek az 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.

Szektor (lemez) (DISC SECTOR)

Szektornak nevezzük a lemez legkisebb olvasható, írható fizikai adategységét.

Szekvenciális hozzáférésű eszköz

L.: Soros hozzáférésű tár

Szelektor (virtuális tárkezelés) (SELECTOR)

Szegmens azonosítója (sorszáma a szegmenstáblázatban).

Személyi számítógép (PERSONAL COMPUTER)

Személyi számítógépnek nevezzük azokat a mikroszámítógépeket, amelyek Intel vagy azzal kompatibilis processzorcsaládra épülnek, és működésük lefelé kompatibilis az IBM-PC hardver- és szoftverelőírásaival. Napjainkban személyi számítógép alatt olyan relatíve kisteljesítményű és viszonylag olcsó számítógépet értünk, amelyet nagy sorozatokban gyártanak; működésük valamilyen Intel vagy azzal kompatibilis processzorra épül; operációs rendszerként a DOS, Windows, Unix stb. különböző verzióinak használata jellemez. A szabványos utasításkészlet miatt a személyi számítógépek szoftverellátottsága széleskörű, ebből fakadóan a legkülönbözőbb alkalmazási területeken - például ipari, mezőgazdasági, egészségügyi, oktatási, irodai, multimédia, hálózati munkaállomás stb. - megtalálhatók.

Szerver= Kiszolgáló (SERVER)

Feladatra orientált, szolgáltató számítógép, amely egy ügyfél-kiszolgáló alapú hálózatban egy konkrét funkciót valósít meg. Például állománykezelés, adatbáziskezelés, levelezés, kommunikációvezérlés, nyomtatás, weblapok kezelése stb. A szerver számítógép a megfelelő szoftverrel több hálózati felhasználó egyidejű kiszolgálására képes.

Szimbolikus cím (SYMBOLIC ADDRESS)

Az adatok, vagy utasítások tárbeli helyének, illetve rekeszének nem annak tényleges helymeghatározó adatával (sorszámával), hanem egy szimbólummal (jelkombinációval) való jelölése. Megjegyzés: A szimbolikus formában történő címmegadás célja a

programozás megkönnyítése. Az algoritmikus gépi nyelvekben a változók azonosítói szimbolikus címek.

Szimmetrikus multiprocesszoros felépítés = SMP (SIMMETRIC MULTIPROCESSORS SYSTEM)

Napjainkban általánosan elterjedt multiprocesszoros architektúra, amelyben azonos, sorozatban gyártott processzorokból épülnek fel a szuperszámítógépek.

Szinaptikus súly

A neurális hálózatokban a neuronok közötti kapcsolat vezérlésére szolgáló érték. Ha egy sejt több, más neurontól kap input értéket, akkor mindegyik input érték egy szinaptikus súllyal lesz megszorozva, és az így képződött szorzatokat adja össze a fogadó neuron. Ezt követően a sejt a kapott értéket egy átviteli függvény szerint átranzszformálja, és ennek eredménye lesz a neuron sejt outputja. A tanulás során a neuronok közötti kapcsolat erőssége, vagyis a szinaptikus súly változik meg egy tanulási szabály alapján, azaz a sejhálózatban a memóriát lényegében a szinaptikus súlyok értékével modellezzük.

Szinkron SRAM = Szinkron sztatikus RAM (SYNCRONES STATIC RANDOM ACCESS MEMORY)

A sztatikus RAM egyik típusa. Olyan ütemezéssel szolgáltatja az adatokat, ahogy azt a processzor várja, azaz a memória működése a processzorhoz szinkronizált.

Szinkron sínvezérlés

A sínrendszeren történő adatátvitel vezérlésének egyik formája. Az eseményeknek rögzített időpontjaik vannak, a sínen kommunikáló eszközök azonos órajellel ütemezettek, az adás-vétel mindig azonos sebességgel történik. Előnye gyorsasága, hátránya viszont, hogy közös órajelet kell biztosítani az összes sínre kapcsolt eszköz számára.

Szinkronizálás (számítógép-részegységek) (SYNCHRONIZATION)

A számítógép részegységei működésének időbeli összehangolása. Ezt általában a gép órajele biztosítja.

Szinkronizáló jel

Egy speciális bitsorozat, aminek feladata az adó és vevő működésének szinkronizálása azaz az órajelképzés időbeli összehangolása.

Szoftver

A számítógépet működőképesé tevő programok összességét (eredeti jelentése: „lágý áru”) nevezzük szoftvernek.

Szoftveres lemezgyorsító tár

A főtár kijelölt része, amely a lemezről kiolvasott adatok átmeneti tárolására szolgál. Ennek kezelése általában az operációs rendszerhez tartozó szoftverrel történik (például DOS esetén SMART DRIVE).

Szolga (SLAVE)

Egy olyan eszköz, amelynek működését egy másik eszköz vezérli. A kommunikációs kapcsolatban a passzív eszközök jelzője. A sínrendszeren megvalósuló adatátvitel során válaszol az adatátviteli igényekre és címekre; sínre teszi illetve fogadja az adatokat.

Sztatikus előrejelzés

A pipeliningban a feltételes vezérlésátadó programutasítások kezelésére alkalmazott spekulatív elágazás-feldolgozásnak egyik módszere. Sztatikus becslésnél a fordítóprogram értékeli ki az ugrási feltételeket, meghatározza a legnagyobb valószínűséggel előforduló ugrási címeket, és ennek megfelelően szervezi a pipelinet.

Sztatikus memória = SRAM (STATIC RANDOM ACCESS MEMORY)

Az írható/olvasható RAM-ok típusa, amely az adatokat félvezető memóriában (flip-flop) tárolja. Ezek tartalma 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. Egy memóriacella helyigénye a dinamikus RAM-okhoz képest lényegesen nagyobb. Gyorsaságuk és magas áruk miatt az SRAM memóriát elsősorban a cachetárolókként használják.

Szubrutin

Olyan gyakran alkalmazott utasítássorozat, amelyet többször felhasználunk a program végrehajtása során, de csak egyszer tárolunk a programmemóriában. Ezeket a programrészeket (rutinokat) szubrutinnak nevezzük.

Szuperskalár processzor

Olyan processzor, amely egy gépi ciklus alatt esetenként több utasítást is képes végrehajtani. Ez azáltal érhető el, hogy a processzor több, párhuzamos működésre képes végrehajtóegységet tartalmaz, és egyidejűleg több pipelinet működtet.

Számítógép (COMPUTER)

Számítógépnek nevezzük azokat az eszközöket, melyek adatok és az ember által megfogalmazott műveletsorozatok (programok) tárolására és automatikus végrehajtására képesek. A számítógép fogalmába beleértjük a működéséhez szükséges programok összességét is.

Számítógép-generáció (COMPUTER GENERATION)

A számítógéprendszerek fejlődését jellemző nagyobb időszak. Az egyes számítógép-generációkat egymástól az alkalmazott alapáramkörök típusa és a műveleti sebesség nagyságrendje szerint különböztetjük meg.

Számítógépház = Ház

Az alaplap és az egyes kártyák és perifériák egy közös fémdobozban kerülnek elhelyezésre, ezt nevezzük számítógépháznak. Többféle típusa létezik, a legelterjedtebb a baby és a mini torony.

Számítógéprendszer architektúra

A számítógép funkcionális felépítése, a részegységek kommunikációs kapcsolatait, valamint a rendszer specifikációjának együttese. Ennek megfelelően egy számítógép architektúrát a részegységek és funkcióik, valamint ezek kapcsolatát meghatározó interfészprotokollok együttes leírásával jellemezhetünk. Az újabb definíciók az architektúra fogalmát egy magasabb absztrakciós szintre terjesztik ki, amely az operációs rendszert is magába foglalja.

Számítógéprendszerek közötti busz (INTERSYSTEM BUS)

Számítógéprendszereket összekötő külső sínrendszer.

Szélessávú összeköttetés

Az adatátvitel rádiófrekvenciás jelekkel, külön csatornákon való megoldása. A szélessávú összeköttetés állomásait koaxiális vagy optikai kábellel kötik össze, így adatot, hangot és képet egyszerre több, frekvenciájuk alapján megkülönböztetett csatornán keresztül vihet át. A szélessávú összeköttetés nagysebességű működésre alkalmas (több mint 20 megabit/másodperc), de drágább, mint az alapsávú, és nehezebb üzembe helyezni. A szélessávú összeköttetés a kábeltelevíziózásban használt technológiára épül.

Színkód (monitor)

A színárnyalatok kódolásához a piros, zöld, kék alapszínek intenzitását kifejező kódérték. Napjainkban egy alapszín intenzitását 8 biten szokták kifejezni, így egy színárnyalat színkódja $3 \times 8 = 24$ bitből áll (True Color színek).

Színmélység (COLOUR DEPTH)

Meghatározza a raszteres képpontok színének kódolásához felhasznált bitek számát. Fontos adata a képernyőknek, szkennereknek, nyomtatóknak. 8 bites színmélység 256 színárnyalat kódolását jelenti, 16 bit 65.536-ot (HiColor) és 24 bit pedig 16,7 millió színárnyalat (TrueColor) kódolását teszi lehetővé.

Szó (WORD)

Olyan rögzített hosszúságú bitsorozat, amely a számítógépes feldolgozás szempontjából egységnek tekinthető. Számítógépekben az adattárolás és adatátvitel igen gyakran szavanként történik. Napjaink jellemző szóhosszai: 8, 16, 32 és 64 bit.

Szórt eszközcímezés

A sínhez csatlakozó eszközök címezésének egyik módja. A master az összes slave megcímezését egyidejűleg (például inicializáláskor) végzi.

Szövegszerkesztő (EDITOR)

L. Editor

Sáv (TRACK)

Olyan nyomvonal az adathordozó felületén, amelynek mentén az adatokat folyamatosan rögzítik a körben forgó mágneslemezre. Megjegyzés: 1. Egy sáv folyamatos olvasáskor vagy íráskor egyetlen olvasó vagy író szerkezeti elemmel áll kapcsolatban. 2. Olyan mozgófejes tárolónál, amelyeknek a tárolóközege lemezköteg, az ugyanazon fejpozícionálással elérhető sávok összegét hengernek nevezik. Egy hengerhez tartozó sávok a forgási tengelytől azonos távolságra vannak.

Sín

L.: Busz

Sínarbitráció

L.: Buszarbitráció

Sínciklus

L.: Buszciklus

Sínfoglalási kérelem engedélyezése

L.: Buszfoglalási kérelmet engedélyező vezérlőjel

Sínfoglalási kérelem

L.: Buszfoglalási kérelem

Sínprotokoll

L.: Buszprotokoll

Sínrendszer

L.: Busz

Sínvezérlés

L.: Buszarbiter

TLB (Translation Lookaside Buffer)

A leggyakrabban használt lapok lapcímfordításhoz szükséges adatait tartalmazó cachetár, amely a processzor és a cachetároló között helyezkedik el.

TLB hit (Translation Lookaside Buffer hit)

Lapozásos virtuális tárkezelésnél az az eset, amikor a virtuális címnek megfelelő lap a főtárban van, és róla bejegyzés található a TLB-ben.

TLB miss (Translation Lookaside Buffer miss)

Lapozásos virtuális tárkezelésnél az az eset, amikor a virtuális címnek megfelelő lapról nem található bejegyzés a TLB-ben.

Ez akkor fordul elő,

- ha a lap a főtárban van, de nincs bejegyzés a TLB-ben (ez a "tisztá" TLB miss);
- ha a lap nincs a főtárban, tehát laphiba keletkezett.

TSS

L.: Taszk állapot szegmens

Tab (billentyű)

(Tabulátor) – szövegszerkesztési billentyű, segítségével a képernyőn egy soron belül nagyobb távolságokat ugorhatunk. A képernyő függőlegesen láthatatlan rácsokra van osztva, a Tab hatására a kurzor a következő rácspontra ugrik. Segítségével balra rendezett oszlopokat hozhatunk létre.

Target

L.: Céleszköz

Task

L.: Taszk

Taszk = Feladat (TASK)

Független folyamatként futtatott önálló alkalmazás vagy alprogram.

Taszk állapot szegmens = TSS (TASK STATE SEGMENT)

Az Intel és vele kompatibilis processzorok védett üzemmódjában taszkváltáskor az egyes taszkok legfontosabb adatai, paraméterei elmentését biztosító szegmens. A hierarchikus privilegizálási szinteken alapuló védelmi rendszer része.

Taszkkapu

Az Intel és vele kompatibilis processzorok tárolóvédelmi rendszerének négy kaputípusa közül az egyik, amely a taszkváltásoknak a védelmi rendszer által ellenőrzött végrehajtását szolgálja. A vezérlésnek kapukon történő "áthaladása" során a privilegizálási szint (PL) mindig ellenőrzésre kerül.

Teljes duplex (FULL DUPLEX)

Olyan összeköttetés, ahol az adatokat egy időben mindkét irányban továbbítani lehet. Ez két vezetékkel tételmez fel, és az adatok továbbítása ekkor mindkét irányban párhuzamosan történik.

Teljesen asszociatív cache (FULLY ASSOCIATIVE CACHE)

Olyan cachetartípus, amelyben az adat memóriacíméből képzett blokksorszám asszociatív módon kerül összehasonlításra a cacheben lévő blokkok címeivel. (Az összehasonlítás rendkívül gyorsan, minden sorra vonatkoztatva azonos időben történik meg.) Egy főtárbeli blokk a cache bármelyik sorába bemásolható, a blokk címe pedig bekerül a cache toldalék részébe. Ez a cachetípus nagyon gyors és egyúttal nagyon drága.

Textúra

Az objektumok felületére ráfeszített mintázat neve a számítógépes grafikában. Segítségével például különböző anyagból álló testek jeleníthetők meg a képen (például márvány, fa textúra).

Time (DOS parancs)

Karbantartó parancs: a gép belső órájának (óra-perc-mp-szmp) kiírása és megváltoztatása.

Timer áramkör

Számlálóáramkör. Az adatátvitel vezérléséhez történő felhasználás esetén feltöltésének megfelelő időtartam elteltét követően megszakítási kérelmet generál.

Tintasugaras nyomtató (INK JET PRINTER)

Olyan lapnyomtató, amely a karaktereket és rajzokat tintacseppecskékből álló sugár kilövésével állítja elő.

Tintasugaras nyomtató (INK JET PRINTER)

Olyan lapnyomtató, amely a karaktereket és rajzokat tintacseppecskékből álló sugár kilövésével állítja elő.

Toldalék (cache) (TAG)

A cache sorának cím- és vezérlési adatokat tartalmazó része.

Részei:

- címrész: főtárból bemásolt blokkra vonatkozó címinformációk;
- vezérlő rész: a cacheblokk adataira vonatkozó érvényességi információk bitenként kódolva.

Trace cache

A P4-es processzor RISC mikroutasításokat tartalmazó L1 cachetárolója.

Track

L.: Sáv

Transaction mode

L.: Tranzakció orientált üzemmód

Transaction

L.: Tranzakció

Translation Lookaside Buffer

L.: TLB

Transport Mode (DMA I/O művelet)

Processzor és DMA között a buszhasználat megosztásának egyik módja. A DMA-vezérlő addig használja a buszt, amíg a processzor nem végez memória-hozzáférést. Ez a buszhasználat megosztási mód a nagygépekre jellemző.

Transzputer (TRANSPUTER)

Egyetlen csipen megvalósított RISC alapú processzor, amelyhez egy memóriamodul is tartozik. Rendelkezik egy DMA-vezérlővel, amely képessé teszi más transzputerekkel való kapcsolattartásra. Ezért megfelelő számú transzputer, "építőköck" szerű összekapcsolásával igen nagy teljesítményre képes szuperszámítógépek építhetők fel. Ez az SMP architektúra első megvalósítása.

Tranzakció (TRANSACTION)

Számítógépes rendszer egy olyan művelete, amely egy igényből és annak kiszolgálásából áll. Ilyen tranzakció például egy lekérdezés egy adatbázisból, vagy egy adatátvitel a számítógép I/O buszán.

Tranzakció orientált I/O

Az input/output igények egyik típusa, amely jellemzője a sok, kis változtatás nagy adathalmazokban (például bankautomaták, repülőgép helyfoglalás). A számítógéprendszer teljesítménye szempontjából a legfontosabb az időegység alatti hozzáférések száma.

Tranzakció orientált üzemmód (TRANSACTION MODE)

Olyan dialógus üzemmód, amikor a felhasználók több munkahelyen, előre jól definiált és azonos jellegű feladatokat végeztetnek el a számítógéppel.

Trapkapu

Az Intel és vele kompatibilis processzorok tárolóvédelmi rendszerének négy kaputípusa közül az egyik, amely trap kivételek ellenőrzött kezelésére szolgál. A vezérlésnek a kapun történő "áthaladása" során a privilegizálási szint (PL) mindig ellenőrzésre kerül.

Tudásalapú architektúra

L.: Neurális hálózat

Tudásalapú hálózat

L.: Neurális hálózat

Type (DOS parancs)

B – állományok tartalmának megtekintése. Általában szövegfájlokra használják.

Tápegység (POWER SUPPLY)

Az elektromos hálózati váltóáramot (220 V) a számítógépek üzemeltetéséhez szükséges alacsony feszültségértékű egyenáramra transzformáló berendezés.

Tár = Tároló, memória (STORAGE, STORAGE DEVICE, STORE, MEMORY)

Minden olyan eszköz, amely adatokat őriz meg későbbi kiolvasás céljára. A tár az elektronikus berendezéseknek az a része, amelyben az adatok és a programok tárolása történik, általában elektromos, mágneses vagy optikai jelrögzítéssel.

Tárgykód (OBJECT CODE)

Forrásprogram fordítóprogrammal való feldolgozásának eredményeként létrejövő program. Bizonyos rendszerekben ez a program már közvetlenül az operatív tárba tölthető és futtatható, más rendszerekben tárgyprogramon a tárba töltéskor még bizonyos műveleteket kell végezni (pl. indirekt címek, külső hivatkozások esetén), illetve egyes rendszerekben, amelyeknél a tárgynyelv egy pszeudó- vagy metanyelv, és a tárgyprogram végrehajtása interpretálással azaz futás közbeni értelmezéssel és a megfelelő gépi kód előállításával történik.

Tárhierarchia = Tárolóhierarchia

A számítógépes rendszerben található tárolók szervezésének elve.
A processzor közelében kis kapacitású, gyors elérésű, drága tárolókat alkalmaznak, és ahogy távolodunk a processzortól, a tárolók hozzáférési ideje, kapacitása nő, valamint az egységnyi adattárolás költsége csökken.
A tárhierarchiába tartoznak a regisztertárak, a cachetárak, a főtár, a háttértárolók és a tömegtárolók.

Tárkezelő egység = MMU (MEMORY MANAGEMENT UNIT)

A virtuális címek fizikai címmé történő leképezését és a címzési hibák felismerését végző hardveregység.

Tárolóhierarchia

L.: Tárhierarchia

Tárolókapacitás

Egy tárolóban elhelyezhető adatok maximális mennyiségét kifejező érték. Leggyakoribb mértékegységei a bájt, kbájt, Mbájt.

Tárolókezelés

Alapfeladata, hogy biztosítsa a processzor műveletvégrehajtásához szükséges adatokat, összehangolva a tárolóhierarchia egyes szintjein lévő memóriaegységek működését.

Tárolóvédelem

L.: Tárvédelem

Társprocesszor = Kooprocesszor (COOPROCESSOR)

Rögzített, konkrét feladatra, önálló működésre alkalmas processzor, amely bővíti az architektúrát a processzor tehermentesítése érdekében. Utasítások, parancsok küldésével "felprogramozható", ezután teljes önállósággal végzi az egyes részfeladatok végrehajtását.

Tárvédelem = Tárolóvédelem (MEMORY PROTECTION, STORAGE PROTECTION)

A taszkok által használt tárterületek védelmét biztosító eljárások összessége, amely kiküszöböli az illetéktelen tárolóhozzáférést illetve módosítást.

Tömegtár

L.: Tömegtároló-egység

Tömegtároló-egység = Tömegtár (BULK STORAGE)

Nagy mennyiségű adat tárolására képes eszköz, (például mágnesszalag, ZIP-lemez, vagy optikai lemez) tároló egység.

Tüzfal szerver

A lokális hálózat és az Internet elválasztására létesített szolgáltatás (a proxy-hoz hasonlóan), azonban célja a biztonság. Szabályozni lehet a belülről kifelé szoló kéréseket, illetve a kívülről befelé folyó forgalmat. [HTTP, FTP, TCP/IP]

UART (Universal Asynchronous Receiver Transmitter)

A számítógép és a perifériális egységek között az aszinkron soros adatátvitelhez szükséges áramkörök, amelyek fő feladata a soros-párhuzamos illetve párhuzamos-soros átalakítás.

UDMA

L.: UltraDMA

USB (Universal Serial Bus)

L.: Általános soros sín

UltraDMA = UDMA

Napjainkban kifejlesztett adatátviteli protokoll az IDE szabvány szerint működő mágneslemezegységek számára, amely a közvetlen tárhozzáféréseken (DMA = Direct Memory Access) alapul. Az UltraDMA javítja az ATA/IDE eszközök átviteli sebességet, és megnöveli az adatátvitel integritását.

Undelete (DOS parancs)

A del-lel letörölt fájlok visszaállítása (ha a háttértáron még megvan és nem copy-val írtunk felül a létező állományt)

Unformat (DOS parancs)

Formattált háttértár előző állapotának visszaállítására szolgáló parancs (format/u esetén nem működik).

Universal Serial Bus

L.: Általános soros sín

Utasítás = Parancs (COMMAND, STATEMENT, INSTRUCTION, ORDER)

- Programnyelv olyan értelmezhető eleme, amely műveletek leírására vagy kijelölésére alkalmas, és e nyelven belül rendszerint önálló egységet képez.
- A számítógép vagy valamely részegysége számára azonnali vagy közvetlen végrehajtást megfogalmazó előírás, amely esetlegesen tartalmazza a végrehajtás körülményeit részletező paramétereket vagy opciókat.

Utasítás-dekódolás

A gépi kódú utasítás-végrehajtás elemi lépései közül az egyik. A műveleti kód és az utasítás-szerkezet értelmezését, az utasításban lévő operandusok fizikai címének kiszámítását hajtja végre ebben a lépésben a processzor.

Utasítás-előkészítés = Utasításle hívás (INSTRUCTION FETCH)

Az utasítás-végrehajtás első elemi lépése. Ebben a fázisban a processzor az utasítás-számláló (PC) tartalma alapján kikeresi a főtárból a programutasítást, és átviszi a vezérlőegység utasítás-regiszterébe (IR).

Utasítás-regiszter (INSTRUCTION REGISTER)

A végrehajtás alatt álló programutasítást tároló regiszter.

Utasítás-szerkezet (INSTRUCTION STRUCTURE)

Az elemi szintű (gépi kódú), a processzor által közvetlenül értelmezhető utasítások felépítése. Ez határozza meg a processzor számára, hogy a gépi utasításokban szereplő bináris értékeket, azaz az utasítás egyes részeit hogyan értelmezze.

Három részből áll (Intel és vele kompatibilis processzorok):

- műveleti kód (opcode), amely meghatározza a processzor számára, hogy milyen műveletet kell végrehajtania;
- operandus hivatkozások (címek), amelyek megadják a processzor számára, hogy hol található az az adat, amelyekkel a műveletet végre kell hajtani;
- kiegészítő, módosító rész, amely olyan kiegészítő információkat tartalmaz a processzor számára, amelyek a műveleti jelrész, illetve az operandus hivatkozások teljes körű (pontos) értelmezéséhez szükségesek.

Utasítás-számláló regiszter (PROGRAM COUNTER, INSTRUCTION POINTER)

A program futása során a legközelebb végrehajtandó utasítás memóriabeli címét (helyét) tartalmazó regiszter.

Utasítás-átrendezés

A pipelining során fellépő problémák kezelésének egyik módszere.

Alkalmazásakor 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.

Utasítás= Parancs(COMMAND, STATEMENT, INSTRUCTION, ORDER)

- Programnyelv olyan értelmezhető eleme, amely műveletek leírására vagy kijelölésére alkalmas, és e nyelven belül rendszerint önálló egységet képez.
- A számítógép vagy valamely részegysége számára azonnali vagy közvetlen végrehajtást megfogalmazó előírás, amely esetlegesen tartalmazza a végrehajtás

körülményeit részletező paramétereket vagy opciókat.

Utasításcache

A RISC processzorokban az utasítások az elődekódoló egységen haladás után, ahol az hozzátész néhány, a dekódolást segítő bitet (az utasítás hosszát) az utasításokhoz, egy igen nagy méretű utasításcachebe kerülnek.

Az AMD K6 utasításcache-e 32 Kbyte (+elődekódoló bitek!), az Intel P55C-nek (Pentium MMX) és a Pentium II-nek (Pentium ProMMX) csak feleekkora utasításcache-e van, a P54C-é (Pentium Classic) pedig csak alig 8Kbyte.

Utasításkészlet

A processzor számára értelmezhető utasítások összessége processzorok egyik fontos jellemzője, ami egy-egy feladatra való alkalmasságukat döntően meghatározza.

Utasításszinten párhuzamos működésű processzor = ILP processzor (INSTRUCTION LEVEL PARALLEL PROCESSOR)

A pipeline utasítás-feldolgozást alkalmazó processzorok gyűjtőfogalma.

Utasításszinten párhuzamos végrehajtás

A pipelining utasítás-végrehajtás jellemzője. Előfeltétele, hogy a gépi utasítás elemi lépéseinek feldolgozását autonóm és párhuzamosan működtethető hardverrendszer hajtsák végre.

Utasítástípus (INSTRUCTION TYPE)

Funkcionális osztályozás szerint az utasításkészletben megtalálható utasítások egy csoportja. A következő főbb utasítástípusokat különböztetjük meg:

- adatátviteli, adatmozgató utasítások;
- műveletvégrehajtó utasítások;
- vezérlő utasítások.

VESA busz (Video Electronics Standards Association Bus)

A mintegy 120, főleg távol-keleti gyártót tömörítő VESA bizottság által kidolgozott 32 bites busz szabvány. A VESA busz saját vezérlőegységgel rendelkezik, órajel sebességű (33 MHz) aszinkron kapcsolatot biztosít 32 bites adatátvitellel. Pentiumoktól kezdve kizsorul a piacról.

VHDL (VHSIC Hardware Description Language)

L.: Hardverleíró nyelv

VLB

L.: VESA busz

VRAM = Képmemória (VIDEO RANDOM ACCESS MEMORY)

A monitorvezérlő kártyákon található DRAM-ok egyik típusa. Jellemzője a kétirányú (dual portos) működés: a VRAM-ok soros hozzáférésű pufferjei a RAMDAC-hoz továbbítják az adatokat. A VRAM a monitorvezérlő kártya grafikus gyorsító processzorával ezzel egyidőben egy másik bemeneti/kimeneti pufferrel tartja a kapcsolatot.

Valós idejű óra = RTC (REAL TIME CLOCK)

A CMOS-ban tárolt időpont és dátum értékét előállító óraáramkör. Ütemezését egy kvarckristály biztosítja.

Valós üzemmód

Az Intel és vele kompatibilis processzorok egyik üzemmódja, amelyben kompatibilitási okokból a processzor 8086-os emulációval működik, azaz azt "hazudja" a felhasználói programnak, hogy a program 8086-os processzoron fut (például 32 bites regisztereknek csak az alsó 8 bitjét engedi használni, az utasításkészlet is ennek megfelelően csökkentett).

Vektorgrafikus modelltér = 3D modelltér

A vektorgrafika modelltere a két- vagy háromdimenziós euklideszi tér. A vektorgrafikában a grafikus objektumokat egy "végtelen" két- vagy háromdimenziós lebegőpontos koordináta-rendszerben ábrázoljuk, kezelésük matematikai eszköze a vektoralgebra.

Vektoros megszakítási eljárás (AUTOVECTOR INTERRUPT)

A hardveres megszakításkezelés egyik formája. A megszakítás-vezérlő egy sorszámot ad át a processzornak, amely a memóriában lévő vektortáblázatban kijelöli a kiszolgáló rutin kezdőcímét.

Vektorszámítógép (VECTOR COMPUTER)

SIMD (egyszeres utasításfolyam, többszörös adatfolyam) architektúrájú számítógép, amely a tudományos-műszaki számítások, szimuláció, 3D grafika stb. területeken gyakran szükséges vektor és mátrix műveleteket gépi utasításszinten hajtja végre.

Ver (DOS parancs)

A DOS verziószámának kiírása.

Verem (stack)

A verem (stack), vagy egy regisztertár a CPU-ban, vagy egy kijelölt memóriaterület a főtárban, melyből az utoljára beírt adatokat lehet először kiolvasni.

Vertikális mikroprogramozás (VERTICAL MICROPROGRAMMING)

A mikroprogramozásnak az az esete, amikor a mikroprogram mikroutasítások sorozataként kerül végrehajtásra. Ehhez a processzor mikroprogram vezérlőegységet, mikroprogramtárat és mikroutasítás számláló regisztert is tartalmaz, és a mikroutasítások műveleti részét a gépi kódú utasításokhoz hasonlóan dekódolni kell.

Vertikális mikroutasítás (VERTICAL MICROINSTRUCTION)

A vertikális mikroprogram egy utasítása, amely a következő mikroutasításnak a mikroprogramtárra hivatkozó címét és műveleti részt tartalmazza. Jellemzője hogy rövid és dekódolni kell.

Veszélyhelyzet

L.: Adatütközés

Vezérlésfüggőség (szuperskalár processzorok)

A pipelineba a végrehajtó egységekhez kibocsátandó utasítások függése attól, hogy egy feltételes vezérlésátadó utasításnál a feltétel kiértékelése után a program hogyan fog elágazni.

Vezérlési mező

A vezérlési pontok beállításához szükséges adatokat bitsorozatként tartalmazó tárolóterület.

Vezérlési pont

A műveleti vezérlés során az adatútvonalak lépésenkénti nyitására és zárására szolgáló kapcsolási helyek.

Vezérlésáramlásos vezérlés (CONTROL FLOW)

A Neumann elvek alapján működő számítógép vezérlésének alapelve, amely szerint a program az utasításokat egymás után, sorban hajtja végre, ettől eltérés csak egy állapotjelző tartalma alapján lehetséges. Ehhez a processzor önálló részegységként vezérlőegységet és utasítás-számláló regisztert tartalmaz.

Vezérlőcsip

L.: Csipkészlet

Vezérlőegység (processzor)= CU (CONTROL UNIT)

Funkcionális egység, amely gondoskodik az utasítások lehívásáról, értelmezéséről és végrehajtásáról. Ennek érdekében előállítja a vezérlőjeleket, amelyek a regiszterek kikapuzását/bekapuzását, az ALU működését és az adatforgalmat irányítják, és előállítja a rendszer működéséhez szükséges állapotinformációt.

A vezérlőegység koordinálja a számítógép valamennyi funkcionális egységének a működését úgy, hogy az események a programnak megfelelő helyes sorrendben és időben következzenek be.

Vezérlőjel (CONTROL SIGNAL)

Belső vagy külső egységek műveleteit vagy kommunikációs protokollok lefutását irányító elektronikus jel.

Vezérlőáramkör-készlet

L.: Csipkészlet

Video ROM-BIOS

A monitorvezérlő kártyán található csak olvasható tár, amely az adott kártya speciális hardver tulajdonságainak megfelelő BIOS-rutinokat tartalmazza.

Video Random Access Memory

L.: Képmemória

Video Read-Only Memory

L.: Video-ROM

Video-BIOS

L.: Video ROM-BIOS

Video-ROM = VROM (VIDEO READ-ONLY MEMORY)

A grafikus kártyán található ROM tár.

Videokártya

L.: Monitorvezérlő kártya

Virtuális cím leképezése fizikai címmé

Az utasítás-végrehajtás során a virtuális címnek megfelelő fizikai főtár cím meghatározása.

Ehhez két dolog szükséges:

- a virtuális tár blokkjainak háttértároló címét, a fizikai memóriába bemásolt blokkok sorszámát és fizikai kezdőcímét megfelelő táblázatokban nyilván kell tartani;
- egy olyan program vagy hardveregység (Memory Management Unit), amely a memóriába bekerült blokkok adatait tartalmazó táblázatok alapján elvégzi a virtuális cím fizikai címmé történő átalakítását.

Virtuális címtartomány = Látszólagos címtartomány

A virtuális címmel elvileg megcímezhető memóriaterület. Nagyságát a virtuális cím felépítése (képzési szabálya) határozza meg.

Virtuális tár = VS (VIRTUAL STORAGE, VIRTUAL MEMORY)

A háttértáron kijelölt tárolóterület (SWAP), amelyik a felhasználó programok számára látszólag egy nagy kapacitású operatív tárként viselkedik. Elsődlegesen többfelhasználós és multiprogramozott rendszerek hatékonyságát növeli. Lehetséges alapeljárások: lapozás, szegmentálás és szegmentálás lapozással.

Visszarendező puffertár = ROB (RE ORDER BUFFER)

A szuperskalár processzorok utasítás-feldolgozása során a soros konzisztencia megőrzése céljából alkalmazott puffertár. Bejegyzései az utasítások állapotának nyilvántartását biztosítják.

Segítségével eldönthető, hogy a processzor egy utasítás eredményadatait mikor írhatja be az átmeneti tárolást biztosító regiszterből az utasításban előírt végleges tárolóhelyre.

Visszaírás (WRITE BACK)

A cacheben megváltoztatott adatok főtárba történő visszaírásának egyik eljárása.

Alkalmazása esetén a cacheben megváltoztatott adatok csak akkor kerülnek visszamásolásra a főtárba, ha a cachének a módosított adatot tartalmazó sorát felül kell írni egy, a főtárból bemásolandó újabb blokkal. A módszer gyorsabb a közvetlen átírásnál, viszont minden cachebeli 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.

Várakoztatás (szuperskalár processzorok)

Az adat- és vezérlésfüggőségeket okozó utasításoknak a végrehajtó egység előtt történő pufferelése.

Védett valós üzemmód

Az Intel és vele kompatibilis processzorok egyik üzemmódja, amelyben a processzor az i8086-os processzort egy taszkban emulálja. (Azaz például a DOS egy WINDOWS alkalmazásként indul el egy ablakban.)

Védett üzemmód (PROTECTED MODE)

Az Intel és vele kompatibilis processzorcsalád multitasking, 32 bites üzemmódja, amelyben a tárolóvédelmi rendszer teljes körűen működik. Ekkor kihasználhatók a 32 bites architektúra összes lehetőségei, ezért a teljesítmény ekkor a legnagyobb.

Végrehajtási jog

Egy taszk jogosultsága egy kódszegmens futtatására.

Végtelen szám

Az IEEE 754-es lebegőpontos számábrázolási szabvány adatformátuma. Ekkor a mantissza 0 értékű, a karakterisztika minden bitje 1-es (mínusz végtelennél az előjelbit 1, plusz végtelennél pedig 0).

WEB szerver (LAN/WAN)

Hipertext (azaz további szövegekre, multimédia elemekre utaló hivatkozást tartalmazó) szövegek továbbítását végző szolgáltató számítógép.

Whestone

Mérnöki, tudományos programokat reprezentáló tesztprogrammal előállított teljesítménymérő mutató

Wideband transmission

L.: Szélessávú összeköttetés

Word

L.: Szó

Workstation

L.: Közepes számítógép

Write back

L.: Visszaírás

Write through

L.: Közvetlen átírás

XT (Extended Technology)

Az 1981-ben bemutatott, legelső IBM PC megnevezése. Jellemző adatai: 8 bites adat, 20 bites címsín.

Z-buffer

A 3D grafikus modelltérben meghatározott, majd a "képernyősíkra" vetített grafikus alakzat látható képpontjainak meghatározására alkalmazott egyik algoritmus a számítógépes grafikában. Nagy erőforrásigénye és a korlátozott alkalmazhatósága (átlátszóság nem valósítható meg stb.) miatt más eljárásokkal kombinálva szokták alkalmazni.

Zsebszámítógép (PALMTOP)

A PC tenyérnyi nagyságú, elemekkel működő változata. Kis beépített képernyője és billentyűzete van. Általában programozható, például Basic nyelven.

gépi utasítás

Egy, a processzor által végrehajtható számítógépes program gépi utasítások sorozatából áll, melyeket bináris formában tárolunk a memóriában.

i80286

L.: 80286

i80386

L.: 80386

i80486

L.: 80486

x86-os emuláció

A processzor mikroprogrammal emulálja az x86-os utasítások végrehajtását a program-kompatibilitás megőrzése érdekében.

x86-os utasításkészlet

Azon utasítások összességének megnevezése, amelyek nem, vagy lényegében csak a regiszter-operandusok hosszában (attól függően, hogy 8, 16, 32 bites processzorról van-e szó) változtak az IBM kompatibilis PC-k megjelenése óta. Ezeket az utasításokat minden PC processzora ismeri, és képes végrehajtani.

Áladatfüggőség (szuperskalár utasítás kibocsátás)

Ha a hivatkozott regiszterek közötti adatfüggőség megszüntethető, például regiszterátnevezéssel vagy az utasítássorrend átrendezésével, akkor áladatfüggőségről beszélünk.

Áladatfüggőségek jöhetnek létre például akkor, ha a programozó rendelkezésére álló regiszterkészlet jóval kisebb annál, mint ahány regiszter a processzorba fizikailag beépítésre kerül (x86 emuláció).

Állapotinformáció

A processzor vagy egy I/O eszköz különböző állapotait azonosító bináris jelzőérték.

Állapotjelző regiszter (processzor) = FLAGS regiszter, EFLAGS regiszter

Ez a regiszter az utasítások végrehajtásának eredményét tükrözi vissza, így tartalmával vezérelhető a feltételes ugróutasítások végrehajtása. Jelző- és vezérlőbitek tartalmaz, amelyeket az aritmetikai-logikai egység állít be. Bitenként kódolásra kerül például, hogy az előzőleg végrehajtott programutasítás eredményadata 0, negatív, történt-e túlcsoordulás stb.

Állapotregiszter (I/O eszközvezérlők)(STATUS REGISTER)

Az I/O eszközvezérlőben ill. a processzorban található egyik regiszter, amely az I/O eszköz ill. a processzor aktuális állapotára vonatkozó információkat tárolja. Például egy merevlemezre egy blokk kiírása megkezdődött, vagy a nyomtatóból kifogyott a papír, egy művelet eredménye negatív stb..

Minden egyes eszközvezérlő és processzor funkcionálisan tartalmazza ezt a feladatot ellátó átmeneti tárolót.

Állomány (fájl, file)

Háttértárolón tárolt adatok együtt kezelt, névvel rendelkező összefüggő csoportja.

Általános célú nyelvek

A hardver orientált nyelvek és a probléma-orientált nyelvek között helyezkednek el, segítségével a feladatok széles spektrumát lehet megoldani. (pl. C, Pascal)

Általános regiszter (GENERAL REGISTER)

Olyan címezhető regiszter, amely a programban különböző célokra is felhasználható.

Általános soros sín = USB, Univerzális Soros Busz (UNIVERSAL SERIAL BUS)

Nagy sebességű soros adatátvitelt biztosító buszszabvány.

Az univerzális soros busz, azaz az USB egy olyan szabványosított csatlakozóaljzat és összeköttetés, amely

- a billentyűzetcsatlakozót,
- az egércsatlakozót,

- a soros és párhuzamos portot

egyetlen nagysebességű, soros átvitelt biztosító összeköttetéssel helyettesíti.

Az USB legfontosabb tulajdonságai a következők:

- 12 Mbit/sec átviteli sebesség,
- maximum 127 USB eszköz kiszolgálása.
- soros adatátvitel,
- lehetővé teszi a plug és play perifériatelepítést,
- az eszközök tápellátása USB kábelén keresztül lehetséges.

Áramkörkészlet

L.: Csipkészlet

Árnyékmemória (SHADOW MEMORY)

A rendszermemóriának az a része, ahova a BIOS eljárásait a rendszerindítás alatt a ROM-ból másolják. Ez fokozza a rendszer teljesítményét, mivel a BIOS eljárások a ROM helyett a több nagyságrenddel gyorsabb RAM memóriában futhatnak le.

Átlapolt memóriakezelés (MEMORY INTERLEAVING)

A memória-hozzáférés gyorsítása a tároló egymástól függetlenül hozzáférhető memóriabankokra történő felosztásával. A tároló 0-ik memóriabankjából kiolvasott adat hozzáférése alatt az 1. memóriabankban lévő következő címen lévő adat már megcímezhető. Ez azt jelenti, hogy címfolytonos olvasás esetén az adatok kiolvasása közel kétszeres sebességgel történhet.

Átnevező puffertár

A regiszterátnevezéshez használt puffertár, amely a regisztereknek a fizikai regiszterekhez történő hozzárendelését valósítja meg.

Átrendező puffertár (RE ORDER BUFFER, ROB)

A processzor a soros konzisztencia megerzéséhez itt tartja nyilván az utasítások állapotát.

Az átrendező puffertár segítségével eldönthető, hogy a processzor egy utasítás eredményadatait mikor írhatja be az átmeneti tárolást biztosító regiszterből az utasításban előírt végleges tárolóhelyre. Ez csak akkor történhet meg, ha a kérdéses utasítást a program eredeti sorrendben megelőző összes utasítás befejeződött (ez azonos azzal, hogy a ROB-ban törlésre került). Ekkor a vizsgált utasítás befejezettnek nyilvánítható és a ROB-ból törölhető (kiírható).

ÉS kapu

AND kapu

Olyan logikai áramkör, amely az ES műveletet valósítja meg, vagyis az adott bemeneti logikai változókra az ES művelet eredményét képezi.

Építőkocka elv (BUILDING BLOCK CONCEPT)

Az elmúlt évek szimmetrikus multiprocesszoros rendszereinek jellemző felépítési elve. Első fizikai megvalósítása a transzputer volt.

Írási jog (tárolóvédelem) (WRITE ACCESS)

Egy taszk felhatalmazása arra, hogy egy másik taszkhoz tartozó tárolóterület adatait átírja.

Órajel = Időzítőjel Clock pulse

Kristályoszillátor által periodikusan kibocsátott impulzus, amely digitális áramkörök működését időzíti.

Órajel-generátor

Egy oszcillátoráramkör, amely a számítógép rendszerbuszának (FSB) órajelét előállítja.

Órajel-sokszorozó = Órajelszorzó

A processzor egy áramköre, amelynek az alaplapi órajel processzoron belüli többszörözése a feladata.

Órajelszorzó

L.: Órajel-sokszorozó

Összeadás (Aritmetikai utasítás) ADD (ADDITION)

Olyan aritmetikai művelet, amely két operandus összegét állítja elő.

Ügyfél-kiszolgáló modell

L.: Kliens-szerver architektúra

Üvegszálas összeköttetés

Számítógépes hálózatokban a fizikai kapcsolat megvalósításának egyik eszköze. Az üvegszál átlátszó anyagból készült vékony szálköteg, amely szaggatott fényjelekkel adatok átvitelére használható. Az üvegszál egyik végén bevezetett fénysugár a szálon belül marad terjedése során, mindig visszaverődve annak belső felületéről.

1. SZÁMÍTÓGÉPEK OSZTÁLYOZÁSA. TELJESÍTŐKÉPESSÉG ÉS SZOLGÁLTATÁSBIZTONSÁG.

1.1. A teljesítőképesség meghatározása

Egy **számítógép architektúrája** alatt az *utasításkészletének felépítését, a számítógép szervezését és a hardveres megvalósítását* értjük. A számítógép-tervezők feladata, hogy olyan architektúrát hozzanak létre, amely eleget tesz a funkcionális követelményeknek, de ugyanakkor megfelel a teljesítőképesség, megbízhatóság, fogyasztás és ár célkitűzéseinek is.

A felhasználási terület függvényében, a különböző számítógép-típusokkal szemben állított követelmények fontossága különböző lehet:

- Az általános célú, otthoni **személyi számítógépek** (*personal computers, PC*) esetében fontos a minél alacsonyabb ár/teljesítőképesség arány, amit olcsóbb processzorokkal, de fejlett grafikai, videó és audió lehetőségekkel érnek el.
- A tudományos és tervezői tevékenységben használt asztali **munkaállomások** (*workstations*) esetében fontos a nagysebességű lebegőpontos feldolgozás és a grafika.
- A kereskedelmi **szerverek** (*servers*) esetében elsődleges az időegység alatt feldolgozott kérések száma, de – a folyamatos igénybevétel miatt – kritikus a magas fokú szolgáltatásbiztonság is. Mivel gyakoriak a szolgáltatások növelésére irányuló igények, ezeknél a gépeknél lehetővé kell tenni a skálázhatóságot, azaz a feldolgozási és tárolókapacitás, valamint az I/O sávszélesség növelését.
- A mindinkább elterjedőben lévő, nem számítástechnikai célú termékekbe **beágyazott számítógépek** (*embedded computers*) esetében fontos a valós idejű működés biztosítása, de minél alacsonyabb áron, ezért nem célszerű a szükségesnél nagyobb teljesítőképességű processzor és nagyobb méretű memória alkalmazása. Ugyanakkor nem elhanyagolható a minél alacsonyabb fogyasztás megvalósítása sem.
- A nagy teljesítőképességű **mainframe-ekre** és **szuperszámítógépekre** (*supercomputers*) jellemző az időegység alatt minél nagyobb mennyiségű – első sorban lebegőpontos – utasítás feldolgozása, amit manapság multiprocesszoros architektúrával, vagy asztali számítógépek klaszterekbe kapcsolásával biztosítanak.

A számítógépek teljesítőképességének (*performance*) mérésénél két mennyiség jöhet számításba:

- **A végrehajtási idő** (*execution time*) – más néven **válaszidő** (*response time*) –, ami egy feladat kezdete és befejezése között eltelt időt jelenti. Ez lehet egy programnak a végrehajtási ideje (másodpercben), vagy egy lemezolvasás ideje (miliszekundumban). A minél kisebb végrehajtási idő az asztali számítógépes-alkalmazások esetében fontos a felhasználónak.
- **Az átadási képesség** (*throughput*) – más néven **sávszélesség** (*bandwidth*) –, ami alatt egy bizonyos idő alatt elvégzett munkát értjük. Ez lehet például egy szerver által másodpercenként feldolgozott tranzakciók száma (tranz./s),

vagy egy lemezegységnél a másodpercenként átvitt megabájtok száma (MB/s). Egy server adminisztrátora a gép minél magasabb átbocsátóképességében érdekelt.

A közelmúlt fejlődését figyelemmel követve azt tapasztaljuk, hogy a sávszélesség növekedése a végrehajtási idő négyzetének növekedésével arányos.

Általában egy A számítógép teljesítőképességét egy másik B gép teljesítőképességével szokás összehasonlítani. Elmondhatjuk, hogy „ A gép teljesítőképessége n -szer nagyobb mint B gépé”, ha

a végrehajtási időt figyelembe véve

$$n = \frac{\text{Teljesítőképesség}_A}{\text{Teljesítőképesség}_B} = \frac{\frac{1}{\text{Végrehajtási idő}_A}}{\frac{1}{\text{Végrehajtási idő}_B}} = \frac{\text{Végrehajtási idő}_B}{\text{Végrehajtási idő}_A},$$

illetve, az átbocsátóképességet figyelembe véve

$$n = \frac{\text{Teljesítőképesség}_A}{\text{Teljesítőképesség}_B} = \frac{\text{Átbocsátóképesség}_A}{\text{Átbocsátóképesség}_B}.$$

A terminológiával kapcsolatban megjegyezzük, hogy a következő kijelentések ekvivalensek:

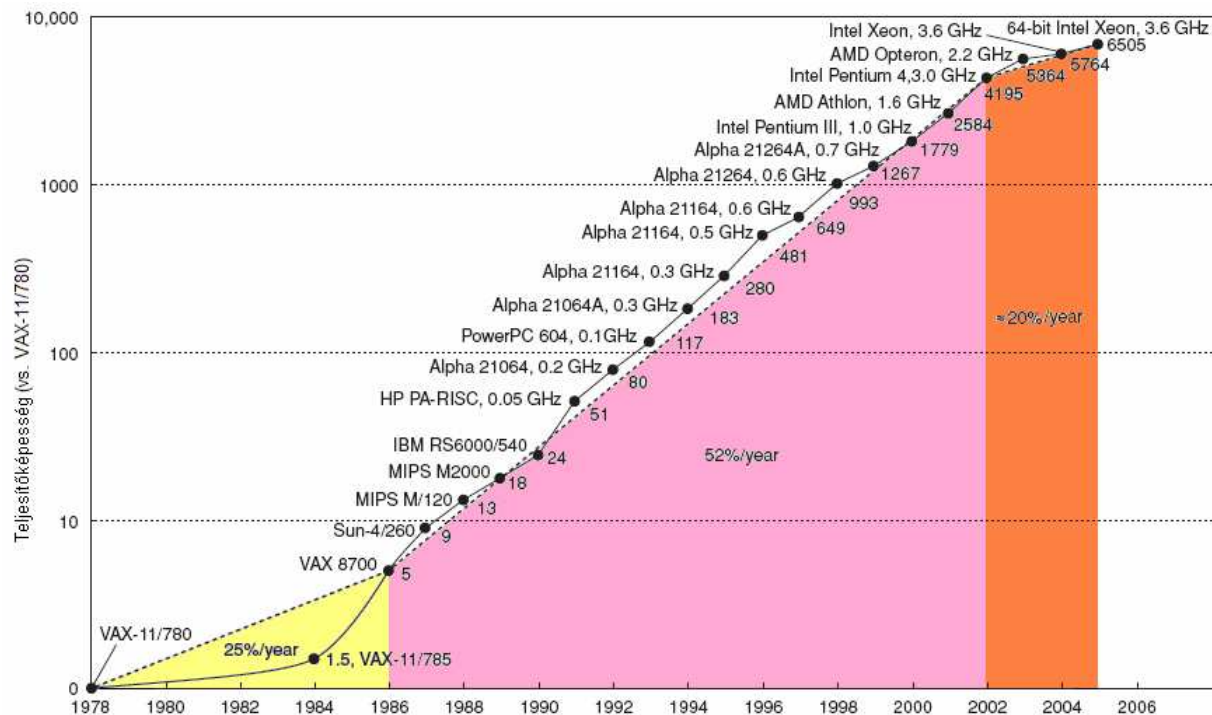
- „ A gép n -szer gyorsabb mint B ” (például 3-szor gyorsabb)
- „ A gép $100(n-1)\%$ -kal gyorsabb mint B ” (például 200%-kal gyorsabb)

1.2. A teljesítőképesség mérése

A számítógépek teljesítőképességét **benchmark** programokkal tudjuk mérni. Ezek olyan programok, amelyeket úgy írtak meg, hogy megpróbálják a valós alkalmazások profilját és viselkedését utánózni. Hogy minél több fajta (például fixpontos, lebegőpontos, grafikus) alkalmazást fedjenek le, a benchmark programokat gyűjteményekben szokták forgalmazni. Íme néhány ismertebb benchmarkgyűjtemény:

- SPEC CPU 2006 (*Standard Performance Evaluation Corporation*) – CPU-teljesítőképességet mérő programok asztali gépek számára. A programok a CPU-időt mérik, igyekezve minimálissá tenni az I/O műveletek okozta idővesztést.
- SPECWeb – több kliens weboldalak iránti kérését szimulálva méri a server átbocsátóképességét.
- TPC (*Transaction Processing Council*) – a server tranzakció-feldolgozó képességét méri, adatbázis-hozzáféréseket és aktualizálásokat szimulálva.
- EEMBC (*EDN Embedded Microprocessor Benchmark Consortium*) – különböző beágyazott alkalmazás (autóipari, távközlési, irodaautomatizálási stb.) teljesítőképességét méri.

A következő ábra a SPECint (fixpontos, azaz *integer*) benchmarkkal mért számítógép-teljesítőképességek időbeli alakulását mutatja be, a DEC VAX-11/780 gépéhez viszonyítva.



Az 1986–2002-es időszakban évi 50%-osnál nagyobb teljesítőképesség-növekedés figyelhető meg, ami nagyrészt a számítógépek architektúráis és szervezési fejlesztéseknek (RISC utasításkészlet, párhuzamosítás, cache) tudható be.

A számítógépek teljesítőképességének kifejezésére hagyományosan szokás a processzor által időegység alatt feldolgozott utasítások számát használni – ez lényegében a processzor átbocsátóképességét fejezi ki (utasítás/s). A gyakorlatban használatos mértékegységek:

- **MIPS (Millions of Instructions per Second)** – a másodpercenként végrehajtott millió utasítás száma, azaz

$$\text{MIPS} = \frac{\text{A program utasításainak száma}}{\text{A program végrehajtási ideje} \times 10^6}$$

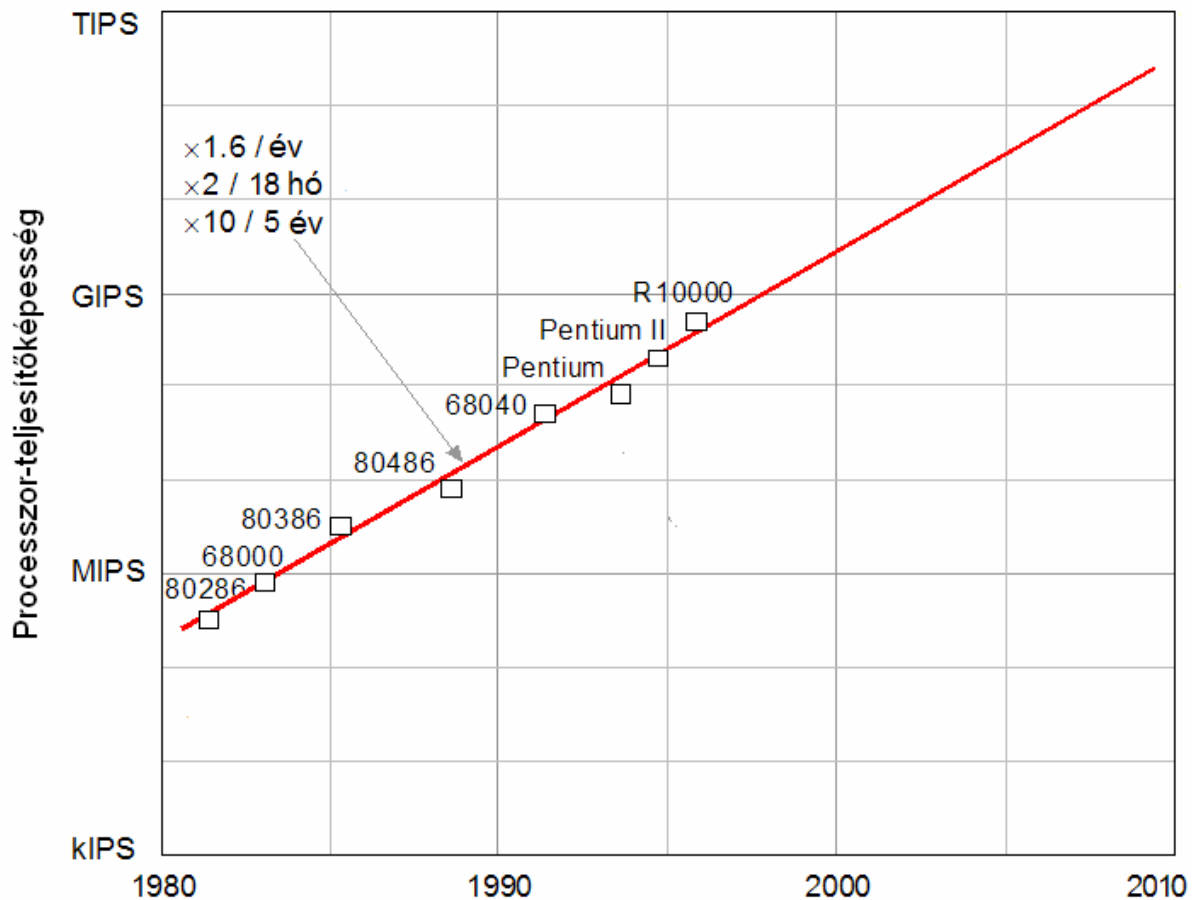
- **MFLOPS (Millions of Floating Point Operations per Second)** – a másodpercenként végrehajtott millió lebegőpontos utasítás száma, azaz

$$\text{MFLOPS} = \frac{\text{A program lebegőpontos utasításainak száma}}{\text{A program végrehajtási ideje} \times 10^6}$$

(Természetesen a MFLOPS mérésére lebegőpontos profilú benchmarkra van szükség.)

A mai nagy teljesítőképességű gépek esetében szokás a mutatók többszörösét (giga, terra) használni, akkor fennáll a GIPS=1000×MIPS, illetve TIPS=1000×MIPS. Az alábbi ábra a processzorok ilyen jellegű

teljesítőképességének az alakulását mutatja be az utóbbi évtizedekben. Látható, hogy a teljesítőképesség kb. ötévenként folyamatosan megtízszereződik, aminek architektúrális, szervezési és technológiai téren tapasztalt haladásra visszavezethető okai vannak.



1.3. A processzor teljesítőképesség-egyenlete

A processzornak ciklikus működése van: egy program feldolgozása a benne lévő utasítások egymás utáni végrehajtását feltételezi, ami időben utasítás-ciklusok sorozatát eredményezi. Feltételezzük, hogy mindegyik utasítás-ciklus a következő öt alciklusokból (gépi ciklusokból) áll:

F (*fetch*) – az utasítás lehívása a memóriából

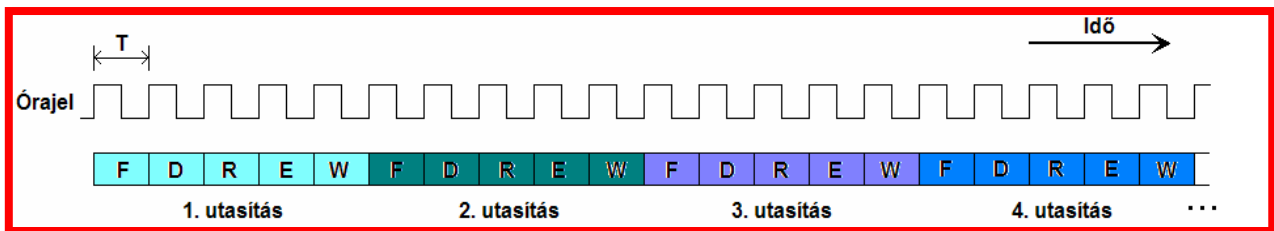
D (*decode*) – az utasítás dekódolása, értelmezése

R (*read*) – az operandus kiolvasása

E (*execute*) – az utasításban megadott művelet végrehajtása

W (*write*) – az eredmény beírása a megadott helyre

Egy-egy utasítás feldolgozása elemi lépések sorozatát jelenti, amelyeknek a végrehajtása időben az **órajel** (*clock*) ütemeinek felel meg. (Az órajel, mint ismeretes, egy periodikus jel, amelynek periódusa a frekvencia fordítottja: $T = 1/f$. Például egy 1 GHz-es frekvenciának 1 ns-os órajel-periódus felel meg.) A hagyományos, Neumann-elvű számítógép processzorának időbeli tevékenységét az alábbi ábra szerint szemléltethetjük:



(Meg kell jegyezni, hogy az egyszerűség kedvéért feltételeztük, hogy minden gépi ciklus egy órajelütem alatt elvégezhető, ami nem mindig így van.)

Az ábra alapján egyszerűen felírható a processzor ún. **teljesítőképesség-egyenlete**, ami egy program végrehajtási idejét (CPU-idő) fejezi ki:

Program végrehajtási ideje = Utasításszám $\times$ Órajelciklus per utasítás $\times$ Órajelciklus ideje

Természetesen a teljesítőképesség annál nagyobb, minél kisebb a végrehajtási idő, mert

$$\text{Teljesítőképesség} = \frac{1}{\text{Végrehajtási idő}}$$

Az órajelciklus per utasítást **CPI**-nek szokás rövidíteni (*Clock Cycles per Instruction*). A rövidítéseket használva, egyenletünk felírható így is:

$$\text{Program végrehajtási ideje} = \text{Utasításszám} \times \text{CPI} \times T$$

Az ábrán lévő példa esetében CPI=5. Feltételezve, hogy az alkalmazásunk végrehajtása 10 000 utasítást jelent, és az órajel 2 GHz-es, a végrehajtáshoz szükséges processzoridő 25 μ s lenne.

A képletben szereplő utasításszám nem a programban lévő utasításoknak a statikus számát jelenti, hanem – a programciklusokat, elágazásokat is figyelembe véve –, a program futása alatt valóban végrehajtott, dinamikus utasításszámot.

Mivel általában nem minden utasítás végrehajtása azonos számú elemi lépésből áll, a CPI értéke is egy átlagot fejez ki:

$$\text{CPI} = \frac{\text{A program összes órajelciklusa}}{\text{Utasításszám}}$$

Figyelembe véve, hogy a programban egy-egy utasítás többször is megtalálható, illetve a végrehajtás közben a ciklusok miatt többször is sorra kerülhet, a program összórajelciklusa kifejezhető mint

$$\text{A program összes órajelciklusa} = \sum_{i=1}^n U_i \times \text{CPI}_i,$$

ahol n a program különböző utasításainak a száma, U_i az i utasítás sorra kerüléseinek a száma, CPI_i pedig az i utasítás órajelciklusainak a száma. Ezzel a CPI kiszámítására a következő képletet kapjuk

$$\text{CPI} = \sum_{i=1}^n \frac{U_i}{\text{Utasításszám}} \times \text{CPI}_i$$

Az egyenletben lévő szorzatnak az első tényezője az i utasítás előfordulásának az arányát fejezi ki a program végrehajtása során.

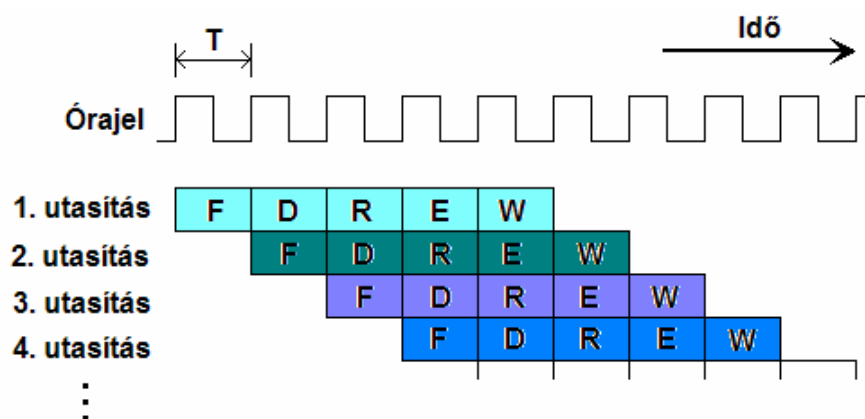
A processzor teljesítőképesség-egyenletéből levonhatjuk a következtetést, hogy a processzoridőben kifejezett teljesítőképesség egyformán függ három tényezőtől, amelyeket csökkenteni kell a teljesítőképesség növelése érdekében:

- *Utasításszám* – a hatékony fordítóprogramtól és az utasításkészlet architektúrájától (pl. CISC, RISC) függ
- *Ciklusszám per utasítás* (CPI) – függ a processzor szervezésétől és az utasításkészlet architektúrájától
- *Órajelciklus ideje* (T) – ennek csökkentése a technológiai fejlődés függvénye

A tervező által kialakítandó számítógép-architektúra szempontjából a CPI csökkentésére kell törekedni. Ennek egyik lehetősége az egyszerű utasításokból (RISC) álló utasításkészlet kidolgozása, amelyeket kevés elemi lépésből végre lehet hajtani – ez viszont másrészt a gépi kódú program utasításainak a számát növelni fogja. A másik lehetőség a tevékenységek **párhuzamosítása**, amelyet különböző szinten lehet megvalósítani:

- *Processzorszintű párhuzamosítás* – ide tartozik a **pipeline** feldolgozásmód, a superskalár, valamint a VLIW-szervezésű processzorok (lásd a Számítógép-architektúrák tárgya I. részét). Mint határesetet, ide sorolhatjuk a mai processzorokban található, SIMD típusú utasítások végrehajtására szakosodott egységet is, amely képes több különböző adaton (vektorelemen) azonos műveletet párhuzamosan végrehajtani.
- *Rendszerszintű párhuzamosítás* – ez a technika több processzor beiktatását jelenti (MIMD típusú **multiprocesszoros rendszer**), amelyek együttműködve hajtják végre az alkalmazás feladatait.

A pipeline szervezésű processzorokra az utasításszintű átlapolt feldolgozásmód jellemző, ami az egész utasítássorozat végrehajtási idejének a csökkenéséhez vezet. Az előző soros utasítás-feldolgozást bemutató ábra pipeline alkalmazása esetén a következőképpen módosul:



Az ábrán látható, hogy a processzor minden órajelütemben befejez (kibocsát) egy utasítást, tehát a CPI=1. Ez ötszörös gyorsulást jelent a soros feldolgozáshoz képest, így a feltételezett példa esetében a végrehajtási idő 5 μ s-ra csökken.

1.4. Amdahl törvénye

A párhuzamosítás eredményeként valamilyen teljesítőképesség-növekedést várunk el, amit **gyorsításban** (*speedup*) fejezünk ki. Ez a feljavított számítógép által nyújtott teljesítőképességet viszonyítja az eredeti gép teljesítőképességéhez, ugyanazon feladat végrehajtásánál:

$$\text{Gyorsítás} = \frac{\text{Teljesítőképesség a feljavított géppel}}{\text{Teljesítőképesség a feljavítás nélkül}}$$

A kérdés az, hogy mennyi lesz ez a teljes gyorsítás, ha a párhuzamosítás által létrehozott gyorsítás a feladat-végrehajtási folyamatnak csak egy töredékét érinti? Erre a kérdésre adja meg a választ **Amdahl törvénye**:

A gyorsabb végrehajtási mód által okozott teljesítőképesség-növekedést korlátozza az az időtöredék, ameddig a gyorsabb módot használni tudjuk. Képletben fogalmazva, az egész feladat végrehajtására vonatkozó teljes gyorsítást a következőképpen számíthatjuk ki:

$$\text{Gyorsítás}_{\text{teljes}} = \frac{1}{(1 - \text{Töredék}_{\text{javított}}) + \frac{\text{Töredék}_{\text{javított}}}{\text{Gyorsítás}_{\text{javított}}}}$$

Ez azt jelenti, hogy például ha a feldolgozás 5%-át nem tudjuk párhuzamosítani, az elérhető gyorsítás nem lehet nagyobb, mint 20, még ha a fennmaradó 95%-ban végtelen gyorsítást is érnénk el.

A képletben a $\text{Töredék}_{\text{javított}}$ azt az időt jelenti az összidőhöz viszonyítva, amelyre a feljavítás vonatkozik. Például, ha az *eredeti* gépen a végrehajtás 8 másodpercet vesz igénybe, és ebből 6 másodpercnyi részt tudunk felgyorsítani, akkor a töredék értéke $6/8=0,75$ lesz. A $\text{Gyorsítás}_{\text{javított}}$ a feljavított részre vonatkozó gyorsítást jelenti, vagyis az eredeti végrehajtási idő arányát a gyorsabb végrehajtási módban elérhető időhöz. Az előző példánál maradva, ha a felgyorsított rész az új gépen 2 másodperc alatt fut le, akkor a gyorsítás mértéke $6/2=3$. Végeredményben az egész feladatra elért teljes gyorsítás:

$$\text{Gyorsítás}_{\text{teljes}} = \frac{1}{0,25 + \frac{0,75}{3}} = \frac{1}{0,5} = 2$$

Belátható, hogy az eredeti 8 másodperc helyett, a felgyorsított gépen a feldolgozás 4 másodpercig fog tartani, tehát a teljes gyorsítás 2 – amennyit kiszámítottunk.

Amdahl törvényét megfogalmazhatjuk egy *általánosított alakban* is. Tegyük fel, hogy egy alkalmazás végrehajtása m szakaszból áll, amelyeknek az f_i aránya ismert az egész végrehajtási időből:

$$f_1 + f_2 + \dots + f_m = 1$$

Ha mindegyik i szakaszt sikerül felgyorsítani p_i tényezővel, akkor a teljes gyorsítást a következő képlet adja meg:

$$\text{Gyorsítás}_{\text{teljes}} = \frac{1}{\frac{f_1}{p_1} + \frac{f_2}{p_2} + \dots + \frac{f_m}{p_m}}$$

1.5. Multiprocesszoros rendszerek teljesítőképessége

A több processzorból álló számítógép-rendszerek teljesítőképességének a kiértékelése komplex feladat, mivel különböző alkalmazások más-más lehetőséget nyújtanak a párhuzamosításra. Ahhoz, hogy a feladatot gyorsabban tudjuk elvégezni, mint az egyprocesszoros gépen, szükséges a programot olyan részfeladatokra bontani, amelyeket szét lehet osztani a processzorok között, majd egyszerre végrehajtani. Feltételezve, hogy a rendszerben n processzor van, a **gyorsítást** (S_n) a következőképpen fejezhetjük ki:

$$S_n = \frac{\text{Végrehajtási idő az egyprocesszoros rendszeren}}{\text{Végrehajtási idő a multiprocesszoros rendszeren}}$$

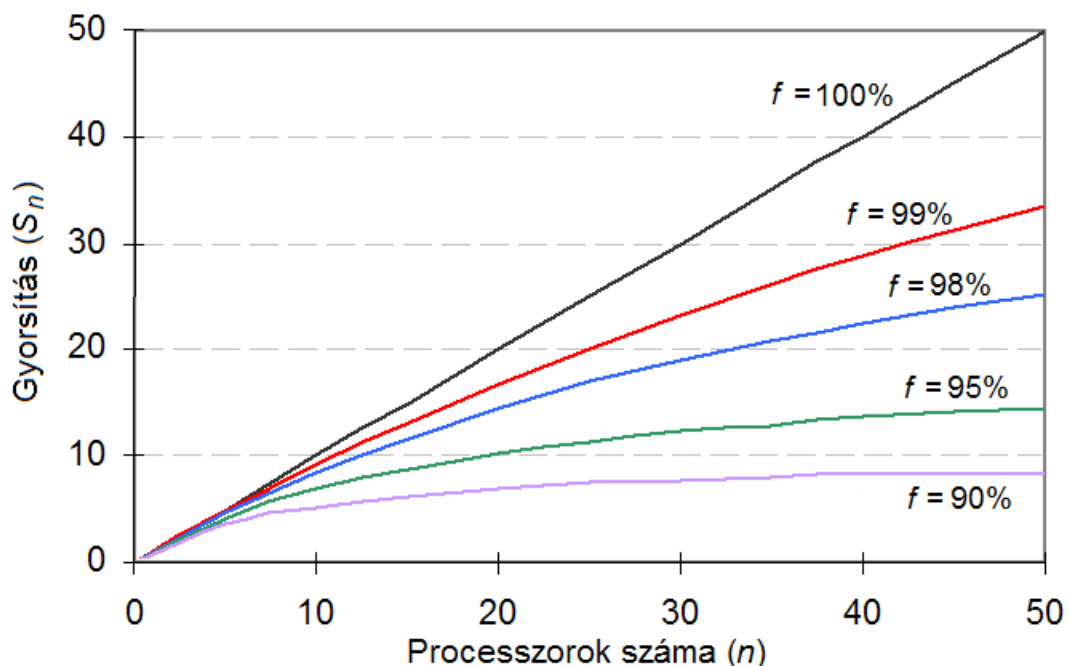
Az ideális esetet az jelentené, ha a programot pontosan n , egyforma végrehajtási idővel rendelkező részre lehet felbontani. Ekkor kapjuk a maximális gyorsítást, amely

$$S_n = \frac{t}{t/n} = n,$$

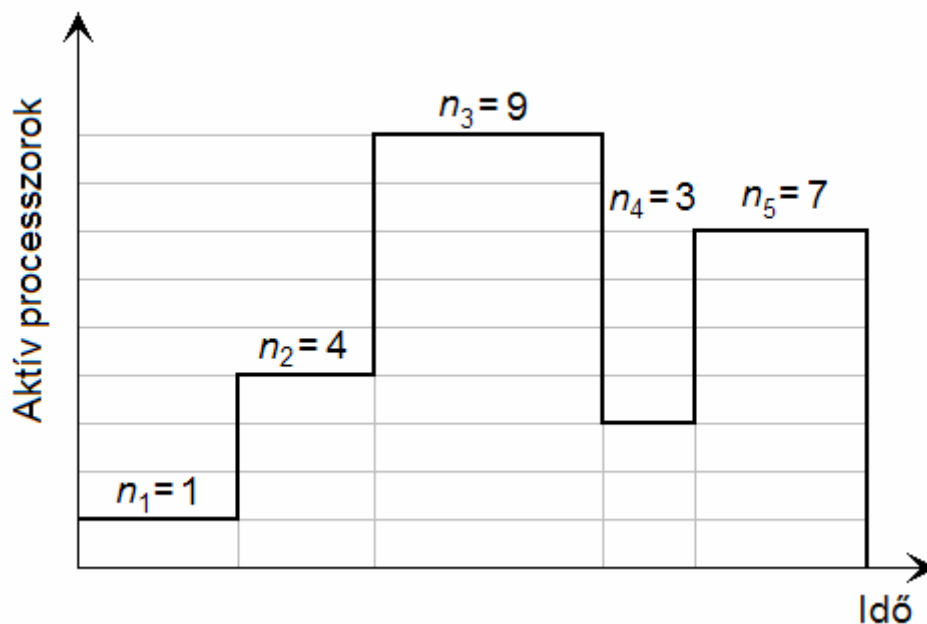
ahol t az egyprocesszoros rendszeren való végrehajtási idő. Ha viszont a számításoknak csak egy részét (legyen ennek az aránya f) lehet párhuzamos feladatokra bontani, akkor az egész felhasználási idő $(1-f) \times t + f \times t/n$ értékre csökken, és a gyorsítási tényezőt az alábbi képlet adja meg:

$$S_n = \frac{t}{(1-f)t + f \frac{t}{n}} = \frac{n}{n(1-f) + f}$$

Ez nem más, mint Amdahl törvénye, amelyben a párhuzamosítás ideje alatt a gyorsítási tényező n . Az alábbi ábra S_n grafikonját mutatja a processzorok számának függvényében. Látható, hogy a multiprocesszoros rendszer alkalmazása a feladat végrehajtásának gyorsítását eredményezi, de hogy ez jelentős legyen, szükséges a párhuzamosítható rész arányának nagyon magas értéke.



A gyakorlatban viszont a párhuzamos feldolgozás időben változó számú processzor igénybevételével történik, amint azt az alábbi ábra mutatja.



Ebben az esetben nincs minden processzor folyamatosan leterhelve, egyesek dolgoznak, mások nem. A gyorsítás kiszámítására Amdahl törvényének általánosított alakját használhatjuk:

$$S_n = \frac{1}{\frac{f_1}{n_1} + \frac{f_2}{n_2} + \dots + \frac{f_m}{n_m}}$$

A képletben n_i az i időintervallumban aktív processzorok számát jelenti, f_i pedig az i intervallum arányát az összydőből.

A valóságban az eddigi elméleti számításoknál rosszabb a helyzet, ugyanis az együttműködő processzoroknak kommunikálniuk kell egymással, ami további időráfordítást jelent. Ez a többletidő nem csak a kommunikációs hálózat és az operációs rendszer kommunikációs szoftvere által okozott késleltetésétől függ, hanem magától az alkalmazói programtól is. Ha a program részfadatait sikerült úgy elosztani a processzorok között, hogy ritkán van szükségük részeredményeket közölni egymásnak, akkor a kommunikációs többletidő kicsi lesz, ellenkező esetben viszont jelentősen megnőhet. Általában feltételezhetjük, hogy a *kommunikációs ráfordítás* aránya függ a kommunikációs hálózat komplexitásától, és így a processzorok számának valamilyen függvénye, mondjuk $c(n)$. Visszatérve a párhuzamosítás idejére egyenletes munkamegosztást feltételező esetre, a kommunikációra fordított többletidő $c(n) \times f \times t/n$ lesz. A gyorsítást a következő egyenlettel számíthatjuk ki:

$$S_n = \frac{t}{(1-f)t + f \frac{t}{n} + c(n)f \frac{t}{n}} = \frac{n}{n(1-f) + f + fc(n)}$$

Ideális esetben, ha az egész feldolgozás párhuzamosítható ($f=1$), valamint a processzorok közötti kommunikációk teljesen átfedhetik egymást, az előbbi egyenletből azt kapjuk, hogy:

$$S_n = \frac{n}{1 + c(n)}$$

A multiprocesszoros rendszer fontos feladata – amint láttuk – a számítások gyorsítása a párhuzamos végrehajtás által. A rendszer **hatékonyságát** (*efficiency*) a következőképpen határozhatjuk meg:

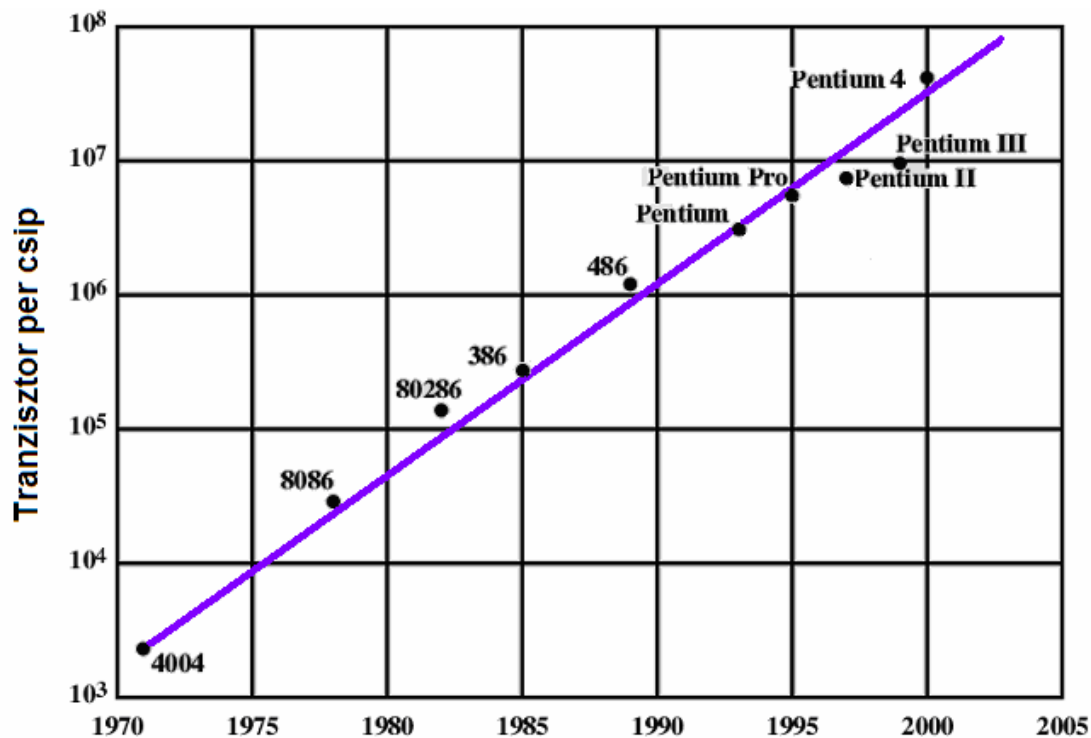
$$E = \frac{S_n}{n} \times 100\%$$

A maximális, 100%-os hatékonyság akkor érhető el, ha n -szeres a gyorsítás, de ez az ideális eset. A gyakorlatban $E < 1$, és a rendszertervezőnek a feladata, hogy a feladatok jó elosztásával a processzorok között, valamint megfelelő multiprocesszoros architektúra kiválasztásával, a hatékonyság minél inkább megközelítse az egységnyi értéket. Mindenképpen el kell kerülni azt az esetet, hogy a kommunikációs ráfordítás domináljon, mert akkor megtörténhet, hogy az egyprocesszoros rendszer hatékonyabb lesz, mint a több processzoros. A rendszer hatékonyságát kedvezően befolyásolja a processzorok minél egyenletesebb terhelése.

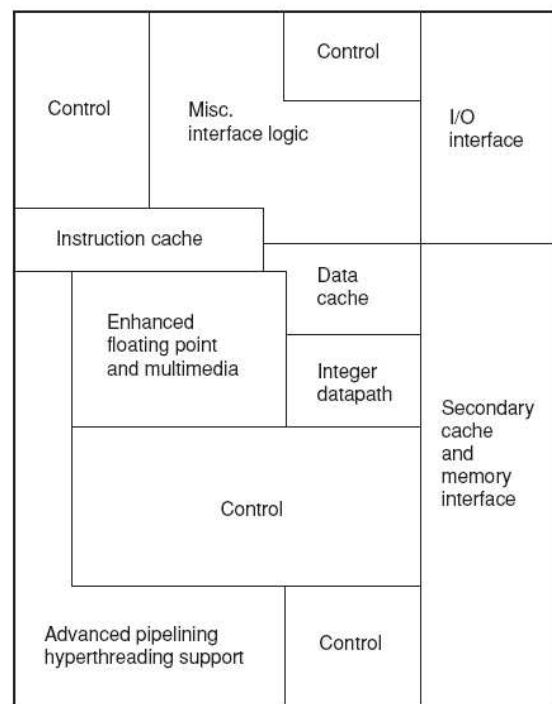
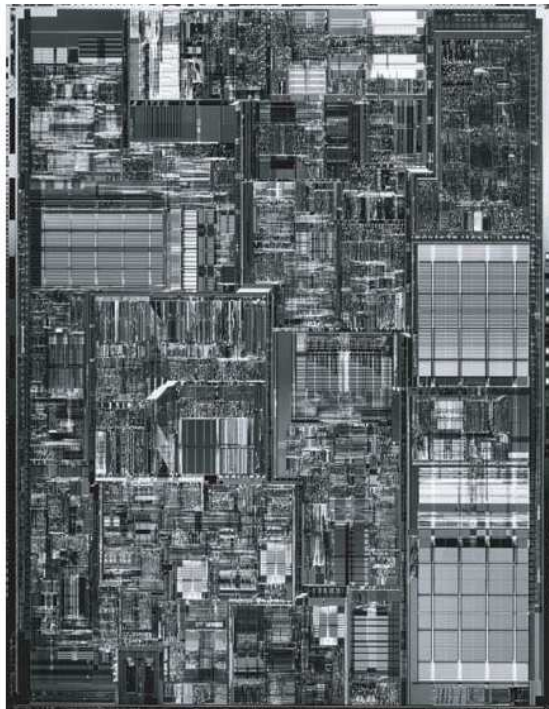
1.6. A technológia szerepe a teljesítőképességben

Az integrált áramkörök gyártási technológiájának meghatározó szerepe van ezek sebességében, méretében, fogyasztásában és árában. Egy processzor gyártójának választania kell a gyorsabb, de kevésbé sűrűn integrálható és nagyobb teljesítményt felvevő *bipoláris* tranzisztor alapú technológia (ECL vagy TTL), illetve a lassúbb, de sűrűbben integrálható és kevesebbet fogyasztó *unipoláris* tranzisztorokat használó technológia (MOS vagy CMOS) között. A legkisebb áramfelvétele a CMOS tranzisztornak van, mivel ez lényegében egy komplementáris tranzisztorpárból áll, amelyből az egyik mindig zárva van. A mai gyakorlatban a processzorgyártók a két technológia kombinációját, az ún. BiCMOS technológiát alkalmazzák, amelyben a nagyobb sebességet igénylő belső részegységeket bipoláris tranzisztorokból, a többit pedig CMOS-ból állítják elő.

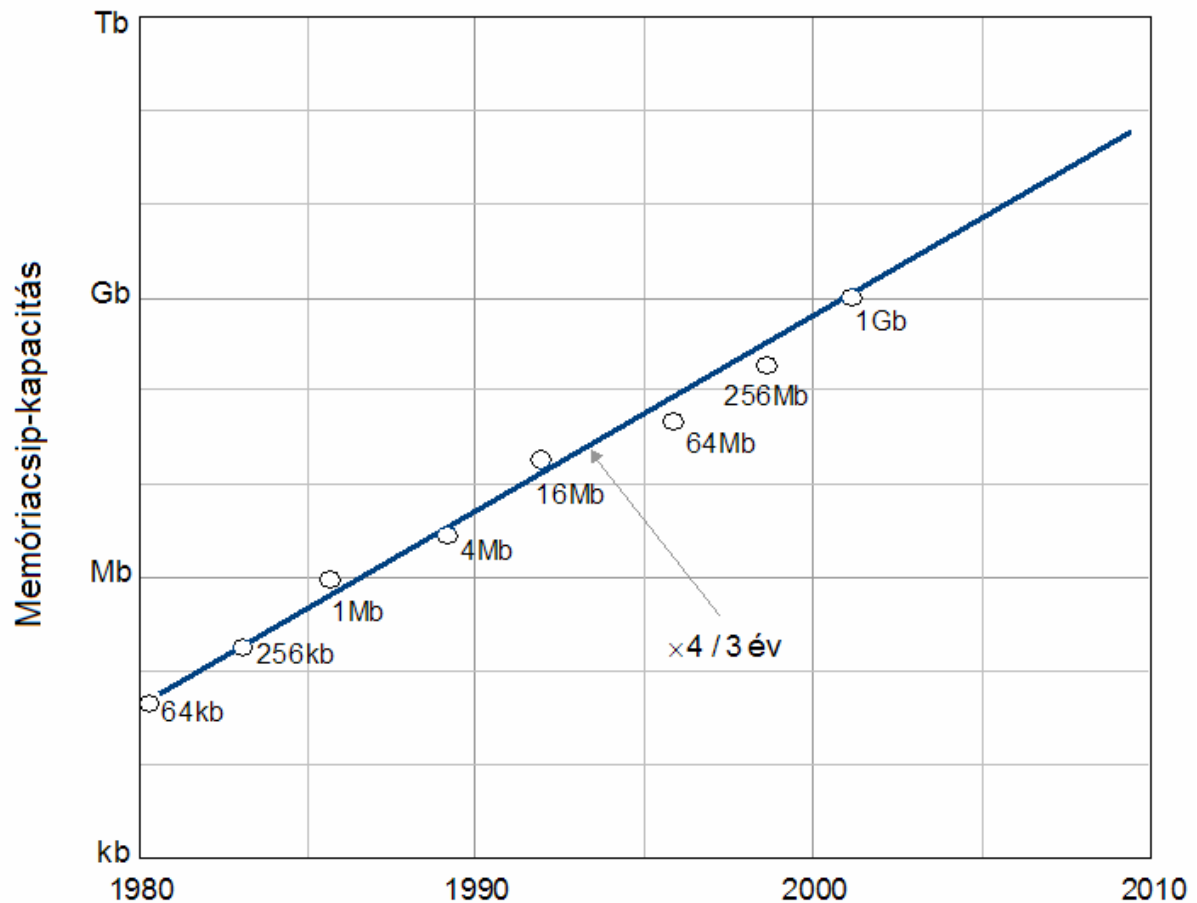
Az integrált áramkörbe lévő tranzisztorok számának növekedési ütemét figyelve fogalmazódott meg 1965-ben a **Moore-féle szabály**: *a csipbe integrálható tranzisztorok száma nagyjából évente megduplázódik, amit manapság inkább úgy érvényesítenek, hogy három évenként megkétszereződik*. Ezt a tendenciát követhetjük figyelemmel az Intel processzorok esetében az alábbi ábrán:



Manapság már **ULSI** (*Ultra Large Scale Integration*) áramkörökről beszélünk, amelyekben az integráltság foka elérte a 100 millió tranzisztort. A magas integráltság azért fontos, mert így egyetlen csipbe sok architektúrális egységet lehet beépíteni, amire szükség van a hyperpipeline-os szuperskalár, esetleg többszálás vagy kétmagos processzoroknál, valamint a 64 bites feldolgozásnál. Például a Pentium 4 Extreme Edition processzor 178 millió tranzisztor tartalmaz. A következő ábra egy Pentium 4-es processzor integrált áramkörét, valamint a belső egységek elhelyezkedését mutatja:



Moore szabályának megfelelően folyamatos növekedést észlelhetünk a DRAM memóriacsipek kapacitásának esetében is (lásd az alábbi ábrát). Ez ugyancsak fontos mutatója egy számítógép teljesítőképességének, és első sorban a technológiai fejlődés eredménye. Az utóbbi években a kapacitás-növekedés lassulni látszik, kb. két évente duplázódik meg.



Íme a korszerű integrált áramkör-technológia néhány fontos jellemzője:

- **Sűrűség.** A csipben lévő tranzisztorok sűrűsége folyamatosan növekszik, mivel a gyártási technológia lehetővé tette a tranzisztorok és vezetékek méretének csökkentését. Egy tengely irányában kifejezve: 130 nm (2002) → 90 nm (2006) → 65 nm (2007) → 22 nm.
- **Sebesség.** A pipeline fokozatainak a késleltetése korlátozza a maximális órajel-frekvenciát. Ez a késleltetés függ egyrészt a logikai kapukat alkotó tranzisztorok kapcsolási sebességétől, másrészt az összekötő huzalok okozta késleltetéstől. A tranzisztorok szempontjából kedvezően hat a sebességre a kisebb méret, de ez nem mondható el a huzalokról is, az egységnyi hosszra eső ellenállás és kapacitás miatt. Ennek következményeként, az órajelciklus mind nagyobb részét kell felhasználni a jelek huzalokon való terjedése okozta késleltetések lefedésére. Az ilyen késleltetések kiegyenlítésére például a Pentium 4-be két plusz fokozatot kellett beiktatni.
- **Fogyasztás.** A tápegységtől felvett villamos teljesítmény kibocsátott hő formájában távozik az integrált áramkörből, és ezt el kell vezetni (ventilátorok és hűtőbordák segítségével). A teljesítmény-felvétel

egyenesen arányos az áramkörben lévő tranzisztorok számával és ezek kapcsolási frekvenciájával. A CMOS csipekben a fogyasztás elsődleges forrása a *dinamikus teljesítmény*, amelyet egy tranzisztornál a következő egyenlet ad meg:

$$\text{Teljesítmény}_{\text{dinamikus}} = \text{Kapacitív terhelés} \times \text{Feszültség}^2 \times \text{Kapcsolási frekvencia}$$

Egy tranzisztor kapacitív terhelése a kimenetére kötött tranzisztorok számának, a tranzisztorok kapacitásának és a huzalok kapacitásának a függvénye. Az első tényezőt a tervrajz, az utóbbi kettőt az alkalmazott technológia határozza meg. A dinamikus teljesítményt a tápfeszültség csökkentésével lehet jelentősen redukálni, ezért ezt az értéket az utóbbi évtizedekben 5 V-ról 1,5 V-ra csökkentették. A kapcsolási frekvencia jelentős tényezővé lépett elő napjainkban a magas órajellel dolgozó processzoroknál. Például egy 3,2 GHz-es Pentium 4 Extreme Edition processzor 135 W-ot fogyaszt, ennek megfelelő hőáramot kell elvezetni a kb. 2 cm²-es csip felületéről, ennek túlmelegedése elkerülésével. Mivel a túlmelegedés veszélye kritikussá vált, a legtöbb processzorban leállítják az éppen inaktív belső egységeknek az órajelét.

A fogyasztás másik összetevője a *statikus teljesítmény*, amelyet a tranzisztorok zárt állapotában is elszivárgó áram okoz:

$$\text{Teljesítmény}_{\text{statikus}} = \text{Áram}_{\text{elszivárgó}} \times \text{Feszültség}$$

Nagyszámú tranzisztort tartalmazó processzorok esetében az elszivárgó áram olyan jelentős, hogy ez felel a fogyasztás egyharmadáért (2007-es adat).

A jövőben az órajel-frekvencia növekedésének a lassulása várható egyrészt a hőkibocsátás korlátai miatt, másrészt, mert a pipeline fokozataira manapság is kevés részfeladat jut, ennek működését nehezen lehet tovább gyorsítani. Egy megoldást jelenthet az egy csipbe beépített több processzormag, amelyek alacsonyabb feszültségen és frekvencián dolgoznak, így csökkentve a kibocsátott hőt.

1.7. A számítógépek szolgáltatásbiztonsága

A **szolgáltatásbiztonság** (*dependability*) egy gyűjtőfogalom, amely a számítógép által nyújtott szolgáltatás iránti bizalmat fejezi ki. A rendszer két állapotban lehet:

- Teljesíti a szolgáltatást a megadott specifikációnak megfelelően
- Megszakítja a szolgáltatást, eltérve a specifikációtól

A szolgáltatás megszakítása a számítógép **meghibásodásának** (*failure*) a következménye, aminek az oka egy alkotóelemben létrejött **hiba** (*fault*). A hiba alkotóelem lehet hardver (például egy memóriacsip), vagy szoftver (például a programozó által tévesen beiktatott utasítás). A hiba latens módon (*latent error*) tartózkodik a gépben, mindaddig, amíg nem aktiválódik, vagyis az általa okozott hiba adat vagy utasítás nem kerül felhasználásra. Amikor ez oda vezet, hogy a számítógép külső viselkedése eltér az elvárttól, akkor beszélünk a gép meghibásodásáról. A hiba kiiktatásával, kijavításával érjük el a számítógép helyreállítását.

Figyelembe véve a két alapállapot, a szolgáltatásbiztonságot több számszerű mennyiséggel jellemezhetjük:

- A **megbízhatóság** (*reliability*) annak a valószínűsége, hogy egy adott időtartam alatt a számítógép ne hibásodjon meg. A gyakorlatban az **átlagos meghibásodási időt** (MTTF – *Mean Time To Failure*) szokták használni:

$$MTTF = \int_0^t R(t) dt$$

A képletben $R(t)$ jelöli a megbízhatóságot, amely mindig $0 \leq R(t) \leq 1$, t pedig az időt. Az MTTF fordítottja a **meghibásodási ráta** (*failure rate*)

$$\lambda = \frac{1}{MTTF}$$

amit meghibásodás/óra mértékegységben fejezünk ki. Több elektronikai alkotóelemből álló, nem redundáns rendszer esetében az egész rendszernek a meghibásodási rátája az alkotóelemek meghibásodási rátájának az összege.

- A **helyreállíthatóság** (*maintainability*) annak a valószínűsége, hogy egy meghibásodott számítógépet adott időtartam alatt meg lehessen javítani. Itt is a gyakorlatban az **átlagos helyreállítási időt** szokás használni, aminek a jelölése MTTR (*Mean Time To Repair*).
- A **rendelkezésre állás** (*availability*) annak a valószínűsége, hogy egy adott időpillanatban a számítógép működőképes legyen, függetlenül attól, hogy előzőleg hányszor romlott el és javították meg. Ennek kifejezésére a gyakorlatban a **rendelkezésre állási tényezőt** szokták használni:

$$K_A = \frac{MTTF}{MTTF + MTTR}$$

A fenti egyenlet nevezője a **meghibásodások közötti átlagos időt** (MTBF – *Mean Time Between Failures*) jelenti:

$$MTBF = MTTF + MTTR$$

Ha megfigyeljük a számítógép eddigi működési–helyreállítási ciklusait, akkor a rendelkezésre állási tényezőt a következőképpen becsülhetjük meg:

$$K_A = \frac{\text{Teljes működési idő}}{\text{Teljes működési idő} + \text{Teljes helyreállítási idő}}$$

TÉTELEK, KÉRDÉSEK

- 1.1. Mi a végrehajtási idő és az átbocsátóképesség? Mondjon példákat az alkalmazásukra.
- 1.2. Mik a benchmark programok? Milyen célú benchmarkokat ismer?
- 1.3. Mit értünk MIPS és MFLOPS alatt?
- 1.4. Adja meg a processzor teljesítőképesség-egyenletét.

- 1.5. Mutassa be ábrán a hagyományos (Neumann-elvű) és pipeline utasítás-feldolgozás elvét.
- 1.6. Fogalmazza meg Amdahl törvényét szavakban és képletben. Mondjon egy példát.
- 1.7. Adja meg és magyarázza el a multiprocesszoros rendszerek gyorsítási tényezőjének egyenletét, ideális és valós esetben.
- 1.8. Jellemezze e korszerű integrált áramköröket sűrűség, sebesség és fogyasztás szempontjából.
- 1.9. Határozza meg a megbízhatóságot, a meghibásodási rátát és a helyreállíthatóságot.
- 1.10. Határozza meg a rendelkezésre állást és adja meg a rendelkezésre állási tényező kiszámításának két képletét.

2. MODELLSZÁMÍTÓGÉP TERVEZÉSI KÉRDÉSEI

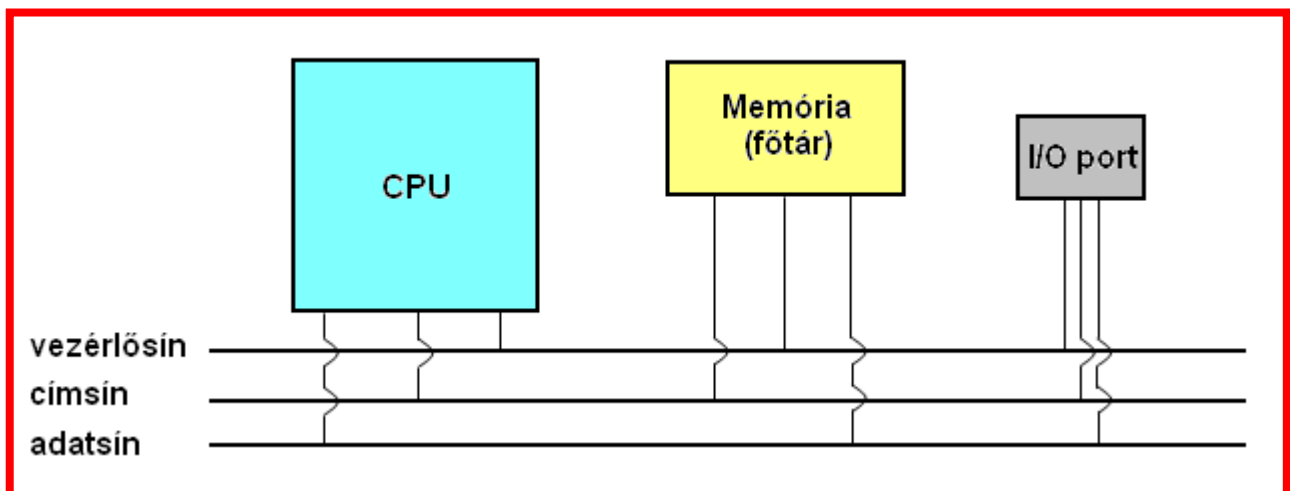
2.1. A modellszámítógép felépítése

[Kovács M., Knapp G., Ágoston Gy., Budai A. – Bevezetés a számítástechnikába, LSI, 2002, VI. fejezete nyomán]

Célunk egy mikroprocesszor-alapú számítógép általános felépítéséből kiindulva ismertetni a jelentősebb tervezési feladatokat, valamint – a következő fejezetben – megtervezni a modellprocesszor utasításkészletét, amely segítségével assembly nyelvű programozási feladatokat tudunk megoldani.

Az elképzelt mikroszámítógépünk a következő részegységekből épül fel (lásd az ábrát):

- **Processzor** (CPU) – lehívja, dekódolja és végrehajtja a memóriában lévő utasításokat;
- **Főtár** vagy **operatív memória** (RAM, ROM) – a végrehajtás alatt álló programokat és ezek adatait tárolja;
- **I/O portok** – a külső eszközökkel (beviteli és kiviteli perifériák, háttértárolók) biztosítják az adatátvitelt;
- **Sínrendszer** – a processzor és a többi részegység közötti kommunikációt szolgálja.



Egy konkrét processzor utasításszerkezetének meghatározásához és az utasításkészletének felépítéséhez azonban ez az általános architektúra önmagában még nem ad elegendő információt. Néhány fontosabb, eldöntendő kérdés:

- Milyen feladatokra akarjuk a regisztereket felhasználni, összesen hány regisztert tartalmazzon a CPU, egy regiszter mérete mekkora legyen (azaz hány bitből álljon)?
- Összesen hány bájtot akarunk megcímezni a számítógép memóriájában, vagy másképp fogalmazva: mekkora legyen a főtár maximális kapacitása bájtban?
- A CPU milyen állapotjellemzőit („flag”-eket) akarjuk rögzíteni az állapotregiszterben?

- Egy lépésben hány bitből álló címeket kell átvinni a címsínen?
- Egy lépésben hány bitből álló adategységeket kell mozgatni az adatsínen a processzor–memória, illetve a processzor és az I/O portok között?
- Milyen puffer és állapotregisztereket tartalmaznak az I/O portok?

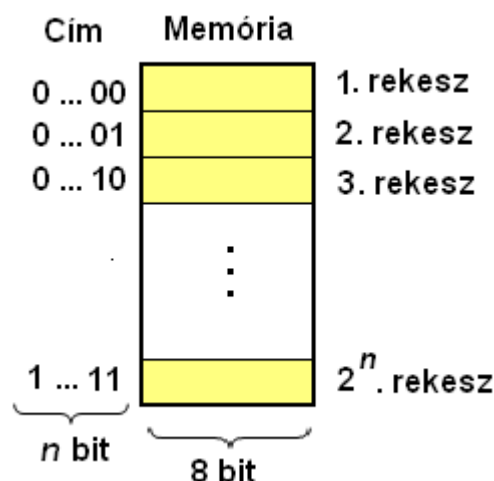
Mint látni fogjuk, ezekre a kérdésekre adható válaszok nem mindig függetlenek egymástól. Az elkövetkezendőkben modellünk konkretizálásával megadjuk a választ az előbbieken feltett összes kérdésre.

Memória

Ha már rögzítettük a főtár maximális kapacitását bájtban, akkor ezzel egyúttal azt is eldöntöttük, hogy

- hány bitesek a címeket tároló regiszterek,
- egy lépésben hány bites címeket kell átvinni a címsínen.

Tegyük fel ugyanis, hogy számítógépünk lehetséges maximális tároló kiépítése esetén a főtár összesen 2^n darab bájtot tartalmaz. Ekkor ezeket a bájtokat n bitet tartalmazó bináris számokkal tudjuk címezni, 0-tól 2^n-1 -ig (lásd az ábrát). Ha viszont egy bájt megcímzéséhez n -bit szükséges, akkor egy lépésben ennyi bitet kell átvinni a címsínen és a címeket tároló regisztereknek is képesnek kell lenniük n -bites számokat befogadni.



A napjainkban használatos Intel processzor alapú számítógépek címsínje egy lépésben 32 bites címek átvitelére képes (32 címvonalat tartalmaz). Ez egyúttal előző gondolatmenetünk alapján azt is jelenti, hogy ezekben a számítógépekben a maximális memóriakapacitás 2^{32} bájt = 4 Gigabájt lehet.

Regiszterkészlet

Egy **processzor regiszterkészletének megtervezése** során egymással ellentétes irányba ható tényezőket kell figyelembe venni:

- Mivel a regiszterek a leggyorsabb tárolók, ezért célszerű, ha a processzor minél több regisztert tartalmaz, mert nyilvánvalóan ekkor lesz a leggyorsabb.
- Minden egyes regiszter a chip áramkörüi egységeiből meghatározott mennyiségűt lefoglal. Emiatt a regiszterek a legdrágább tárolóeszközök, így a sok regiszter processzor árát is jelentősen megnöveli.

Erre figyelemmel a processzorok tervezése során gondosan mérlegelni kell, hogy milyen feladatokra hány darab regisztert építünk be a CPU-ba.

Az elkövetkezendőkben modellprocesszorunk regiszterkészletéből az **általános regisztereket** fogjuk definiálni, azaz azokat a regisztereket, melyeket a gépi utasítások közvetlenül felhasználhatnak. Elméletileg egy processzor már akkor is működőképes, ha csak három regisztert, egy akkumulátort, egy utasításszámlálót és egy állapotregisztert tartalmaz. Ebben az esetben viszont például a verem címének kezelése, vagy a tömbök feldolgozásához szükséges indexelés csak a memóriában lenne megoldható. Ez nagyon bonyolulttá tenné a programokat és a processzort is lelassítaná. (A regiszterek hozzáférési ideje kb. tizedrésze a memóriának.)

Erre is figyelemmel a modell **processzorunk regiszterkészletét** a következőkben határozzuk meg:

AX Akkumulátor (*Accumulator*)
BX Bázisregiszter (*Basis register*)
CX Számláló regiszter (*Counter register*)
DX Adatregiszter (*Data register*)
SP Veremmutató (*Stack pointer*)
BP Bázismutató (*Basis pointer*)
SI S-Indexregiszter (*Source index*)
DI D-Indexregiszter (*Destination index*)
IP Utasításszámláló (*Instruction pointer*)
FLAGS Állapotjelző

Itt az AX, BX, CX, DX, SP, BP, SI, DI, IP a regiszterek szimbolikus rövid nevét jelöli. Az egyes regiszterek felhasználását a programokban a későbbiek során fogjuk bemutatni.

Az egyszerűség és a jobb szemléltethetőség érdekében a modellünkben szereplő **regiszterek hosszát egységesen 16 bit-ben határozzuk meg**. Ha 16-bites regisztereink vannak, akkor célszerű biztosítani a memóriából **16-bites (2 bájtos) szavak kiolvasását illetve visszairását**. Ebből néhány dolog azonnal következik:

- A processzor maximum 16-bites, illetve 15-bites számokkal képes előjel nélküli, illetve előjeles műveleteket elvégezni;
- Számítógépünk maximális memóriakapacitása 2^{16} bájt = 64 Kilobájt;
- Számítógépünk címsínje 16 címvonalat tartalmaz;
- Számítógépünk adatsínje 16 adatvonalat tartalmaz.

Feltételezzük, hogy a főtár memóriavezérlő áramkörei a szavas hozzáférés mellett az **1 bájtos memóriaműveleteket** is lehetővé teszik. Ebben az esetben a sínrendszer úgy viszi át, a processzor pedig úgy fogadja a 8-bites bájtokat, hogy azok a regiszterek alsó legkisebb helyi értékű 8 bit-jén kerülnek letárolásra. Ezeket a bájtokat az AX, BX, CX, DX regiszterekben AL, BL, CL, DL-lel fogjuk jelölni, és az 1 bájtos memóriaműveleteket csak e regiszterekre engedjük meg.

I/O portok

Mikroszámítógép modellünk nagyon hiányos lenne, ha nem biztosítaná az **input/output műveletek** végrehajthatóságát. Ahhoz, hogy az utasításkészletben input/output utasításokat is szerepeltetni tudjunk, konkrétan specifikálni kell azokat az I/O portokat, melyekhez a perifériák kapcsolódhatnak. Modellünkben **két I/O portot** definiálunk.

- *0-s port*, mely input adatok beolvasására szolgál,
- *1-es port*, mely output adatok továbbítására szolgál.

Az egyszerűség kedvéért képzeljük azt, hogy a 0-s port a billentyűzetről érkező adatokat fogadja, az 1-es port pedig az adatokat a monitor képernyőjére történő kiíráshoz továbbítja.

Minden port modellünkben **két 8-bites regisztert tartalmaz**. Ezek:

- *a pufferregiszter*, mely az adatok átmeneti tárolására szolgál;
- *az állapotregiszter*, mely a port állapotinformációit tárolja.

A következőkben megadjuk az állapotregiszterek egyes bitjeinek értelmezését:

A 0-s port állapotregiszterében:

2^0 bit = 1, ha a port pufferregiszterébe új adat érkezett az input eszköztől

= 0, a port pufferregisztere változatlan, azaz nem érkezett új adat

2^1 bit = 1, ha a port pufferregiszterét a CPU még nem olvasta ki

= 0, ha a port pufferregiszterét a CPU már kiolvasta

Az 1-es port állapotregiszterében

2^0 bit = 1, ha nincs az output eszközhöz továbbítandó adat az 1-es port pufferében

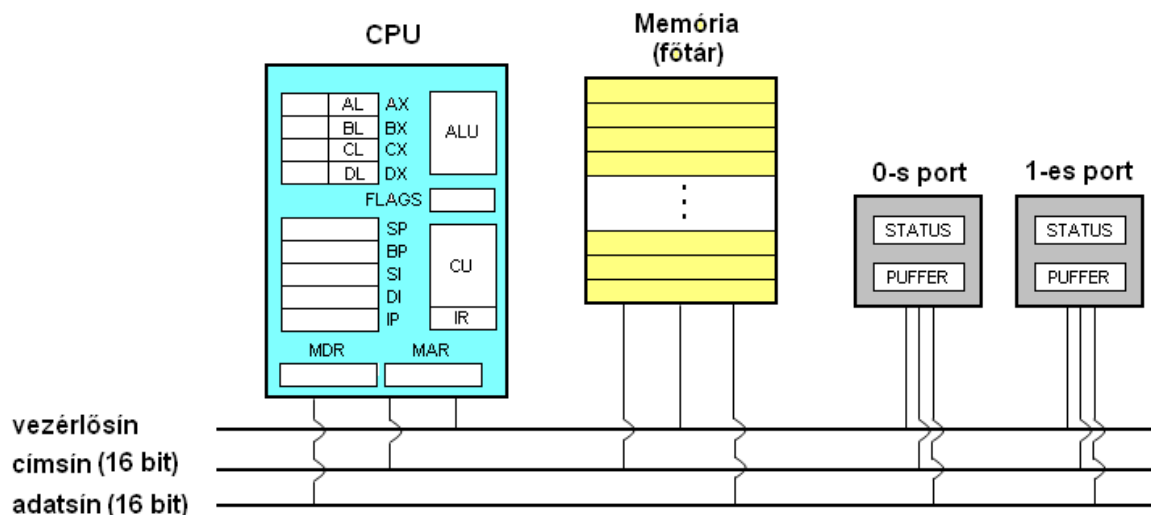
= 0, ha az output eszközhöz továbbítandó új adat van az 1-es port pufferében

2^1 bit = 1, ha az output eszközhöz való adattovábbítás rendben befejeződött, a port felkészült új adat fogadására

= 0, ha az output eszközhöz való adattovábbítás még nem fejeződött be

Az állapotregiszterek 0-s értékű bitjeit a CPU, az 1-es értékű bitjeit pedig a külső eszközök állítják be. Nyilvánvaló, hogy az input/output művelet végrehajthatóságának a CPU oldaláról az a feltétele, hogy az állapotregiszter 2^0 és 2^1 helyi értékű bitjei 1-es értékűek legyenek.

Az előző specifikációk alapján most már felvázolhatjuk **modellszámítógépünk pontos felépítését**:



A korábbiakban tárgyalt regisztereken kívül az ábrán még három **rendszerregisztert** is feltüntettünk. Ezek:

- az IR utasításregiszter (*Instruction Register*), mely a memóriából kiolvasott programutasítás átmeneti tárolására szolgál a CPU-ban
- az MAR memória címregiszter (*Memory Address Register*), mely a memóriából kiolvasandó adatok címét tárolja;
- az MDR memória adatregiszter (*Memory Data Register*), mely a memóriába beírandó, illetve onnan kiolvasandó adatok átmeneti tárolója.

Mivel 1, illetve 2 bájtos adategységeket akarunk mozgatni a CPU és a főtar között, nyilvánvaló, hogy MDR-nek 16-bites regiszternek kell lennie. Címeink 16 bitesek, ezért MAR is 16-bites regiszter. IR méretét nyilván az határozza meg, hogy mekkora egy gépi utasítás maximális hossza. Mint ezt a következő fejezetben le fogjuk vezetni, ez 32 bit, ennek megfelelően IR a modellünkben legyen 32-bites regiszter.

Az IR, MAR, MDR rendszerregiszterek, melyek egy gépi utasítás végrehajtásán belüli elemi lépésekben játszanak fontos szerepet. Ezek a programutasításokkal nem elérhetők, nem címezhetők és tartalmuk sem változtatható meg.

2.2. A buszciklusok

A program utasításainak a feldolgozása során a processzornak gyakran kell használnia a külső sínrendszert adatátvitel érdekében. Ezek az úgynevezett **buszciklusok** (*bus cycles*) sok utasítás végrehajtásához szükségesek, és hasonló módon zajlanak. Ez lehetővé teszi a processzor vezérlő egységének egyszerűbb megtervezését, mert azonos műveletsorozatot kell végrehajtania minden utasításnál ugyanannak a típusú buszciklusnak az esetében. A legfontosabb buszciklusok a következők:

- utasítás kiolvasása (lehívása) a memóriából (*fetch cycle*);
- olvasás a memóriából (*read cycle*);
- írás a memóriába (*write cycle*);
- bevitel (olvasás) egy port regiszteréből (*in cycle*);

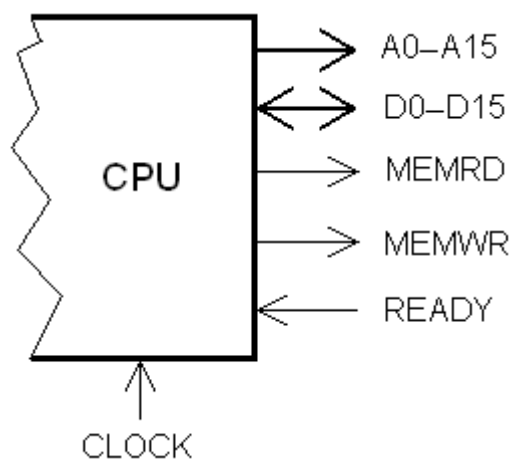
- kivitel (írás) egy port regiszterébe (*out cycle*);
- megszakítási ciklus (*interrupt cycle*).

Az első három ciklus a leggyakoribb, ezek az alapciklusok. Az utasításlehívás is megegyezik a külső lebonyolítása szempontjából egy memória-olvasással, de a processzoron belül két sajátossággal rendelkezik:

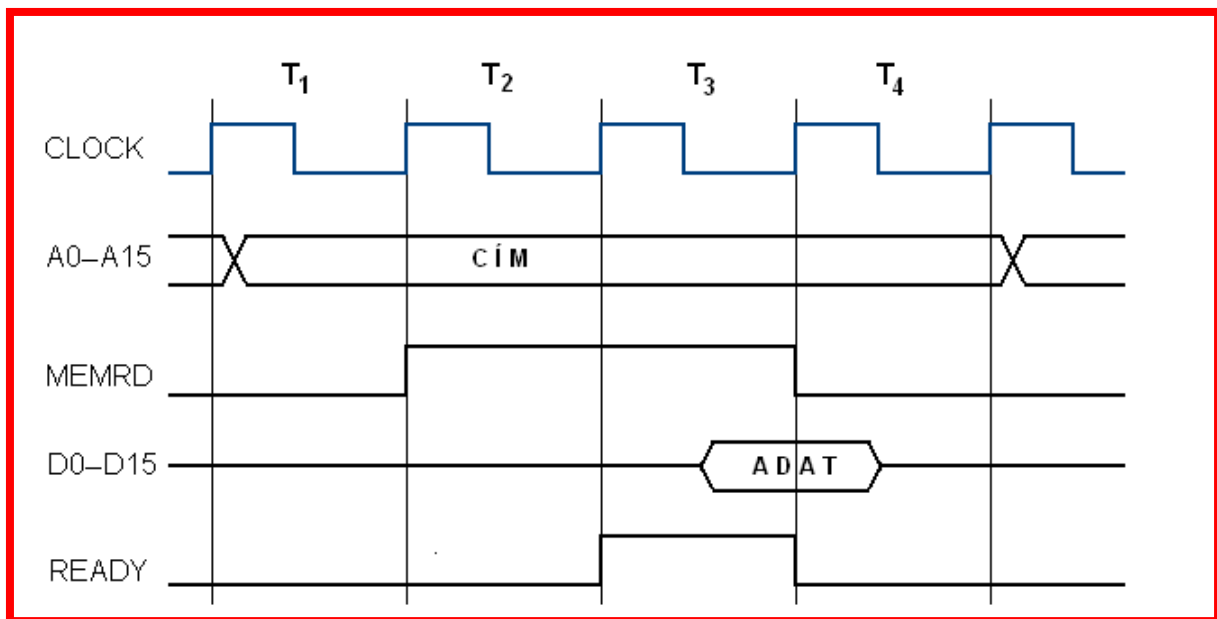
- az olvasás az utasításszámlálóban (IP) lévő címről történik;
- a beolvasott adat, ami ebben az esetben az utasítás kódja, az utasításregiszterbe (IR) kerül.

Egy **memória-olvasási ciklus** a processzor külső buszrendszere szintjén a következő tevékenységeket feltételezi (lásd az ábrát is):

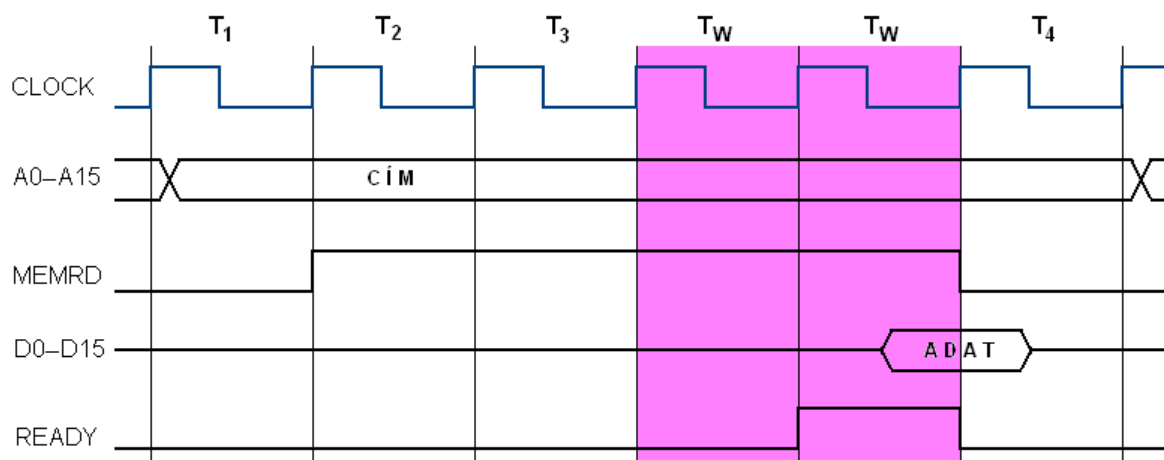
1. Az olvasni kívánt memóriarekesz címének kihelyezése a címbuszra (A0–A15).
2. A memóriából való olvasást jelző vezérlőjel (MEMRD) aktiválása.
3. Várakozó állapotok beiktatása (a READ bemenet „0”-ra állításával) ha szükséges, abban az esetben ha a memória túl lassú a processzorhoz képest.
4. A memóriából érkező adat beolvasása az adatbuszról (D0–D15)



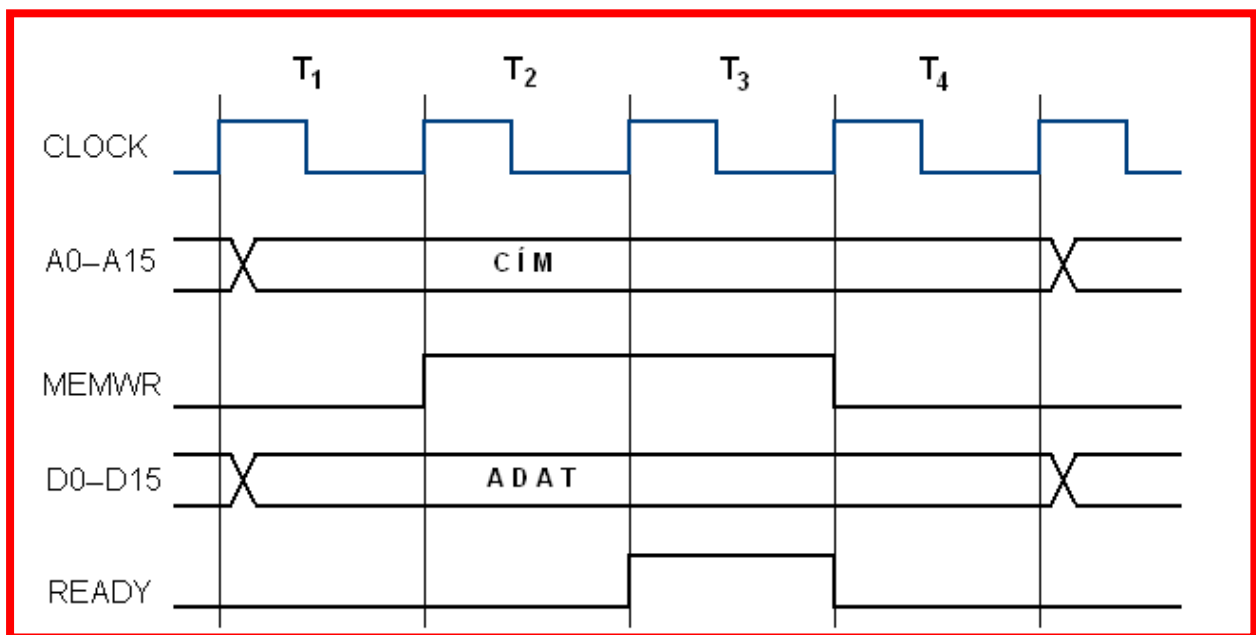
Mindezeket a műveleteket a processzor az órajelhez szinkronizálva hajtja végre, amint az a következő idődiagramból is látszik:



Egy olvasási ciklusban a kiolvasott szót a memóriának idejében az adatbuszra kell helyeznie, hogy azt a processzor helyesen átvehesse. Abban pillanatban, amikor a processzor mintavételezi az adatbuszt (a mi esetünkben a T₄ órajelciklus kezdetén), az adatvonalaknak már stabil állapotban kell lenniük. Ha viszont ez nem lehetséges, mert a memória-áramköröknek a hozzáférési ideje nagyobb, mint a válaszra rendelkezésre álló órajel-periódus, akkor T_w várakozó állapotot (vagy állapotokat) kell beiktatni a processzor buszciklusába. Ezt a READY bemenet megfelelő ideig tartó alacsony szintre („0”-ra) hozásával lehet megtenni. A következő ábra az olvasási ciklust két várakozó állapot beiktatásával mutatja:



Az írási ciklus az olvasásnak a fordítottja, ekkor a processzor helyezi az adatot az adatbuszra, majd a memóriába írást vezérlő jelet (MEMWR) aktiválja:

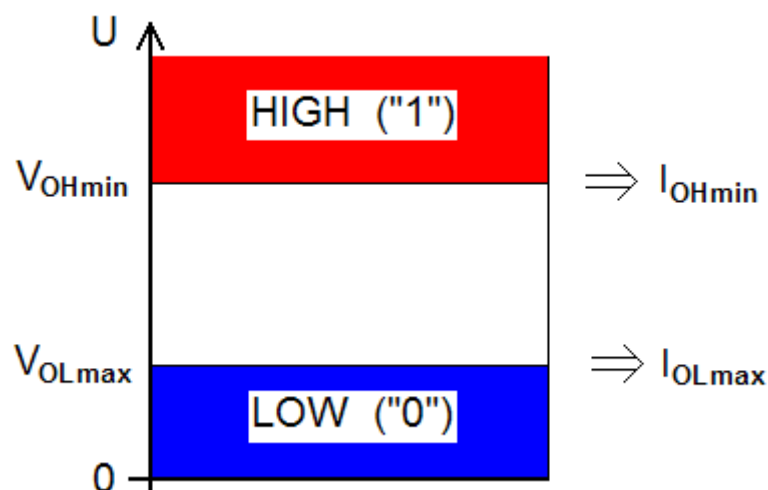


A kiviteli-beviteli buszciklusok lefolyása megegyezik a memóriaciklusokéval, azzal a különbséggel, hogy a processzor az I/O részegységekre vonatkozó vezérlőjeleket aktiválja (IORD, illetve IOWR).

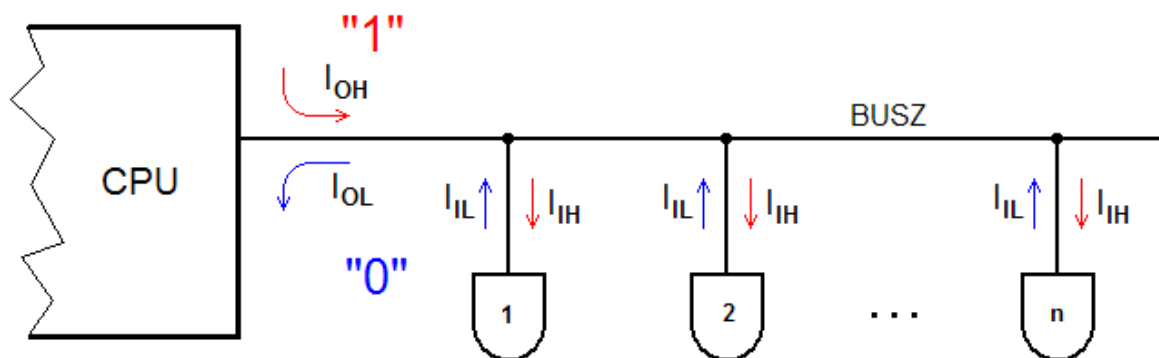
A ciklusok idődiagramjait a processzor gyártója mindig megadja, mert csak ennek betartásával lehet a rendszer többi egységét a buszhoz csatlakoztatni.

2.3. Külső sínek kialakítása

A külső busz teszi lehetővé a kapcsolatot a processzor és a többi részegység között. A bináris adatok helyes átvitele érdekében szükséges, hogy a buszvonalak feszültségszintjei a logikai „0”, valamint „1” értékeknek megfelelő intervallumban állapodjanak meg:



Mivel egy buszvonatra több vezérelt kapu van kapcsolva, Kirchhoff első törvényének megfelelően, a processzor kimenetének biztosítania kell a kapuk által igényelt áramok összegét, úgy az alacsony (*low*, azaz „0”), mint a magas (*high*, azaz „1”) állapotban:



(Az ábrán látható, hogy a „0” állapotban a processzor felveszi az áramot, az „1”-es állapotban pedig kiadja.) Minden logikai áramkör, így **a processzor kimenetének is van egy névleges (katalógusban megadott) áramerőssége, amit biztosítani tud.** Ezt az értéket a két feszültségtartomány szélső értékeire (V_{OLmax} és V_{OHmin}) adják meg, ami az első ábra alapján I_{OLmax} és I_{OHmin} . **Feltételezve, hogy a processzor egy kimenetének a buszon keresztül n darab – a részegységekben lévő – kaput kell vezérelnie, a helyes logikai szintek biztosítása érdekében az alábbi feltételeknek kell teljesülniük:**

$$I_{OLmax} \geq \sum_{1}^n I_{ILmax}$$

$$I_{OHmin} \geq \sum_{1}^n I_{IHmin}$$

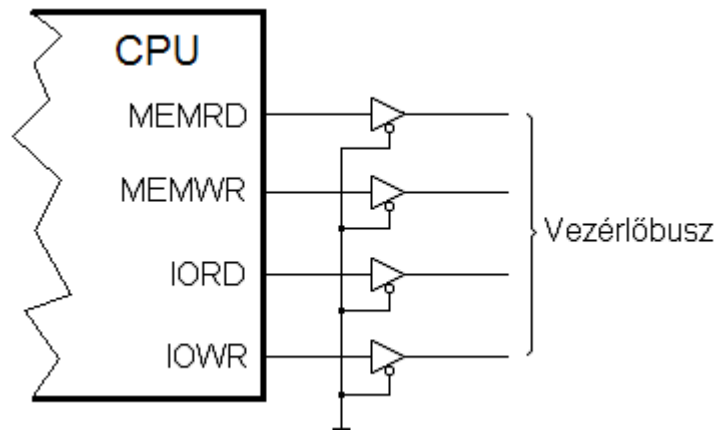
A használt jelölések a következők:

- O – output (kimenet)
- I – input (bemenet)
- L – low (alacsony szint)
- H – high (magas szint)
- max – a V_{OLmax} határértéknek megfelelő
- min – a V_{OHmin} határértéknek megfelelő

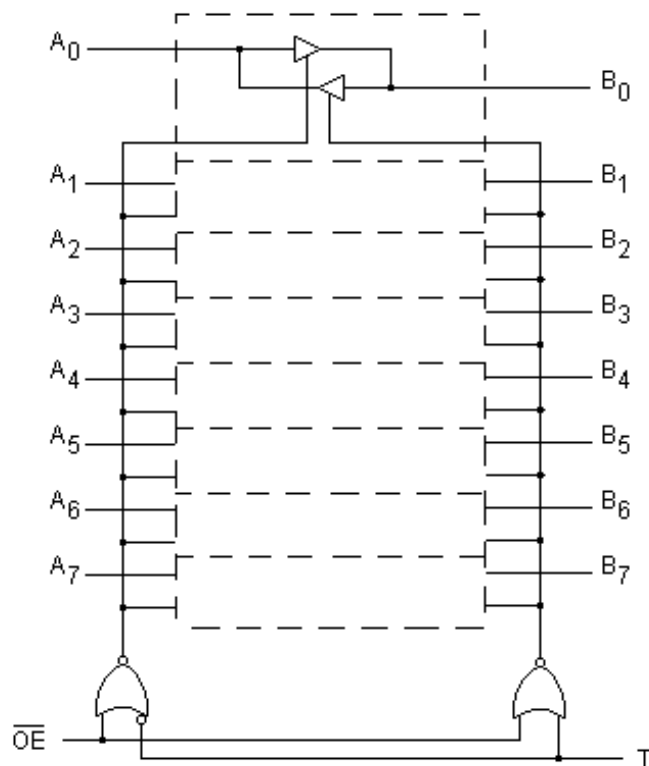
Az előző képleteket egyszerűbben tudjuk alkalmazni, ha az áramokat egy tipikus áramkörcsalád standard bemeneti áramaihoz viszonyítjuk. Ekkor a kimeneti, illetve bemeneti **terhelhetőségi tényezőkkel** (FO – *fan-out*, FI – *fan-in*) dolgozhatunk:

$$FO \geq \sum_{1}^n FI$$

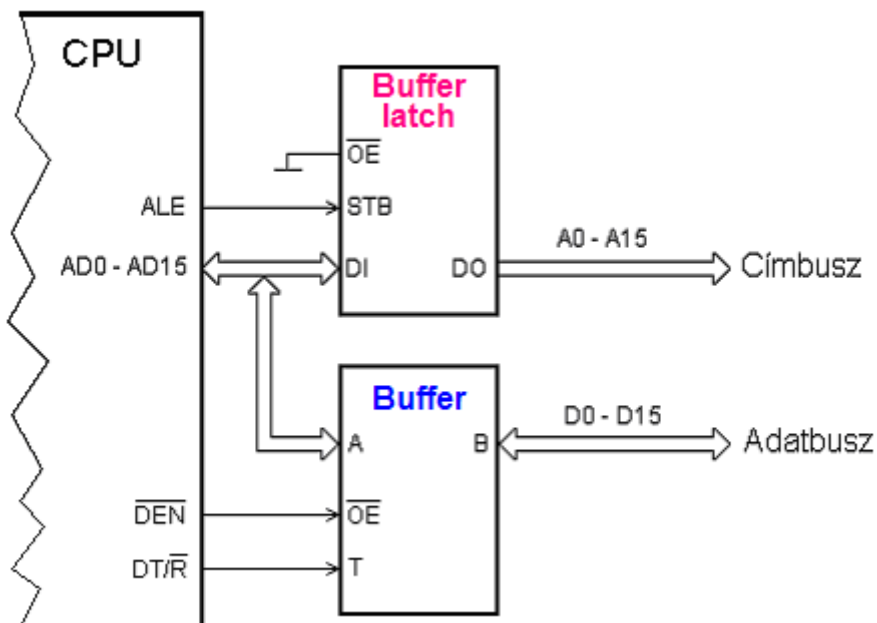
Általában a mikroprocesszorok kimeneteinek kicsi a terhelhetőségi tényezője, ezért áramerősítést végző **meghajtó (driver)** kapukat kell beiktatni a processzor kimenete és a rendszersín közé. Célszerű ezt a meghajtót **háromállapotú kapukból (tri-state)** kialakítani, mert így a harmadik, magas impedanciájú állapotban át lehet engedni a buszvezérlést például egy DMA-vezérlőnek. Egy ilyen áramkör alkalmazására mutat példát a következő ábra, a vezérlőbusz kialakítása esetében:



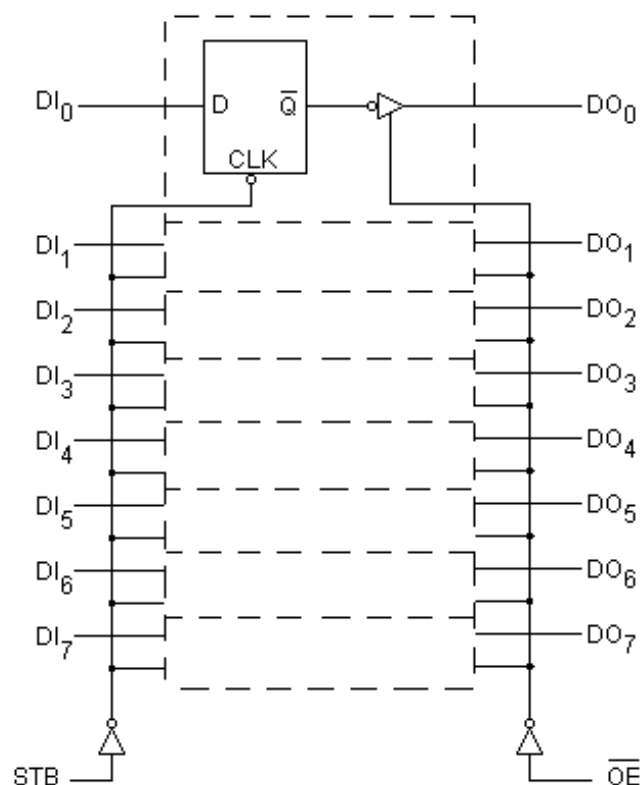
Mivel az adatbusz kétirányú, itt kétirányú **puffer meghajtókra** (*buffer driver*) van szükség. Egy ilyen 8 bites áramkörre mutat példát az alábbi ábra (OE = *Output Enable* ha aktív, megnyitja a kimeneteket, T = *Transmit* az adatátvitel irányát határozza meg):



Sok processzor **multiplexált cím- és adatbuszt** használ, így csökkentve a tok lábacskáinak a számát. Ebben az esetben a közös buszt demultiplexálni kell, azaz a különálló buszokat a processzoron kívül kell kialakítani:



Minden buszciklus (írás vagy olvasás) a cím kiadásával kezdődik. Multiplexált busz esetében ezt a címet egy külső regiszterben kell eltárolni, hogy a ciklus további részében is a címbuszon megmaradjon. A tárolás az ALE (*Address Latch Enable*) impulzussal történik. A használt *buffer latch* áramkör belső kapcsolási rajzát az alábbi ábra mutatja:



Amikor a közös buszon az adatok haladnak, a processzor egy DEN (*Data Enable*) jelet aktivál – ezzel nyitjuk meg az adatpuffert. Az adatátvitel irányát a DT/R (*Data Transmit/Receive*) jel vezérli.

Természetesen a buszvonalak megfelelő vezérlését meg kell oldani minden olyan részegység esetében is, amely adatokat küld a buszon (memória, interfész). Ezeknek a buszra kapcsolódó kimeneteit is három állapotú puffer meghajtókkal kell ellátni úgy, hogy ha ez egység nem küld adatot, a kimenetei magas impedanciájú állapotban legyenek. Ekkor egy másik egység használhatja a buszt, elkerülve az adatok ütközését.

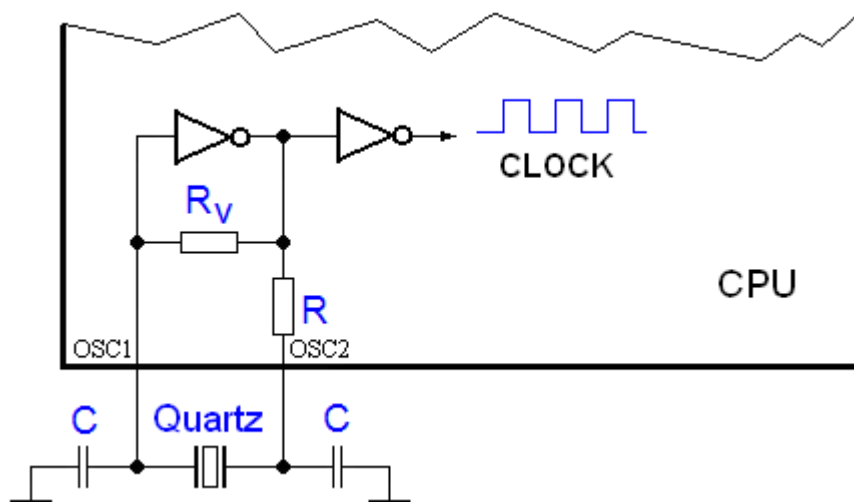
2.4. Az órajel-generátor

Az órajel (*clock*) alkalmazására minden processzor esetében szükség van, mert ez biztosítja a belső áramkörök szekvenciális működését. A hőmérséklet- és feszültségváltozásokra is stabil frekvencia biztosítása érdekében egy **kristályoszillátort** használnak, mivel a kvarckristály a piezoelektromos tulajdonsága miatt a sajátrezgését nagyon pontosan tartja. A kristály úgy viselkedik, mint egy RLC-áramkör (ellenállás-induktivitás-kapacitás), amelynek a rezgési frekvenciája a méretétől és alakjától függ.

Az processzor típusától függően, az órajel-generátornak két kialakítási módja terjedt el:

- külső áramkör, amelynek a jelét a processzor megfelelő (CLOCK) bemenetére kell alkalmazni
- a processzorba beépített oszcillátor, amelyre kívülről kell rákapcsolni a kvarckristályt (ezt a konstrukciót tipikusan a mikrokontrollereknél használják)

Az utóbbi megoldásra mutat példát a következő ábra:



Elektromos feszültség hatására a kristály deformálódik, majd a feszültség megszűnése után visszaveszi az eredeti alakját, és ő generál feszültséget. A rezgést az oszcillátor úgy tartja fenn, hogy az áramkör a kvarctól vett jelet felerősíti, és az inverter kimenetéről visszajuttatja a kristálynak. Amikor az így létrejött váltakozó feszültség megegyezik a kristály saját frekvenciájával, akkor rezonál. A második kapu csak erősítési célokat szolgál, a meghajtott további kapuk számára. Az ábrán látható R , C paramétereket a következő egyenletből számítjuk ki (C értéke kicsi, általában 10-30 pF-os):

$$f = \frac{1}{2\pi RC}$$

Ebben f a frekvenciát jelöli.

A processzor névleges működési frekvenciáját a műszaki adatai között találjuk meg, de figyelmet kell fordítani az órajel kitöltési tényezőjére, valamint az impulzusok élleinek időtartamára is.

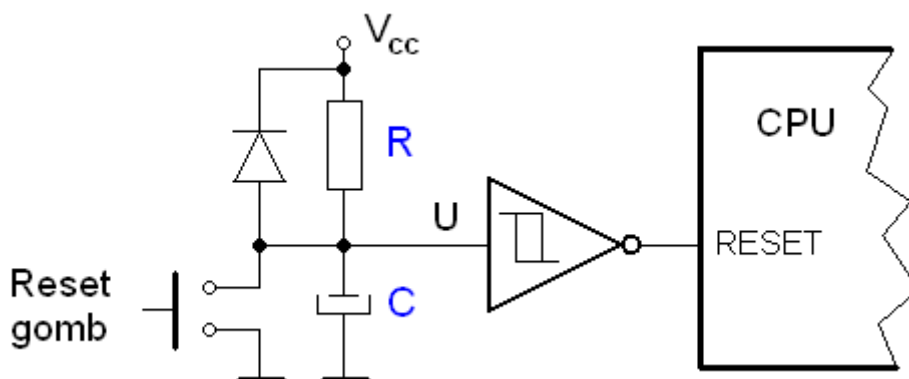
2.5. A RESET áramkör

Minden processzor rendelkezik egy RESET bemenettel, amely lehetővé teszi a processzor inicializálását, azaz kezdeti állapotba hozását. Erre egyrészt a számítógép indításánál van szükség, másrészt akkor, ha a kezelő a *Reset gomb* megnyomásával újra akarja indítani a gépet.

A RESET impulzus alkalmazásának általában két hatása van a mikroprocesszorban:

- az utasításszámláló kezdőértékre állítása (ez legtöbb processzornál nulla);
- a maszkolható megszakítások elfogadásának letiltása (ez mindaddig fennmarad, amíg a futó program egy engedélyező utasítással a tiltást fel nem oldja).

Egy RESET áramkör egyszerű kapcsolási rajzát mutatja az alábbi ábra:

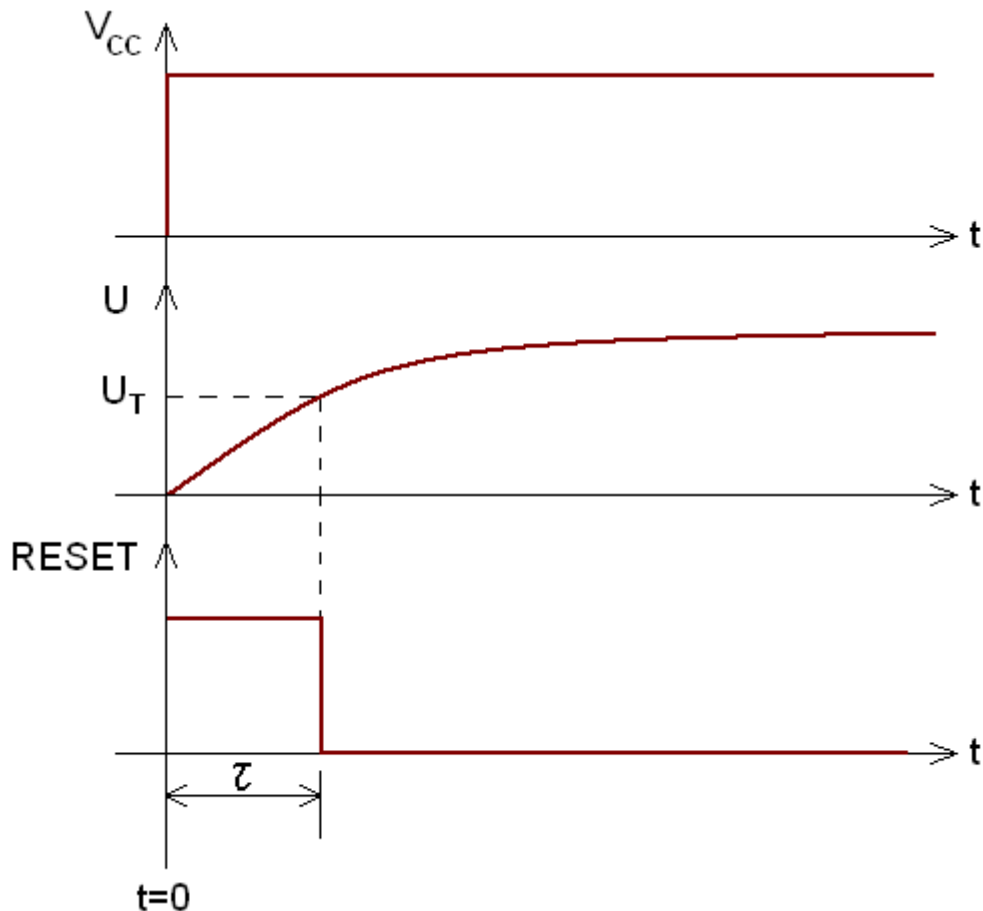


A RESET gomb megnyomására egyértelmű, hogy az inverteren keresztül pozitív impulzus kerül a processzor bemenetére, ami ezt alaphelyzetbe állítja. Az R-C páros a gép bekapcsolásánál biztosítja az automatikus inicializálást. Az inverter egy *Schmitt-trigger*, amely pontos küszöbfeszültségen (legyen ez U_T) kapcsolja át a kimenetet logikai „1”-ből „0”-ba. A tápfeszültség (V_{CC}) megjelenésekor, a kondenzátor elkezd exponenciálisan töltődni az ellenálláson keresztül:

$$U = V_{CC} \left(1 - e^{-\frac{t}{RC}} \right)$$

Amíg ez a feszültség el nem éri az U_T értéket, az inverter kimenetén logikai „1” van, ami alaphelyzetbe hozza a processzort. A küszöbfeszültség elérésének pillanata jelenti a pozitív impulzus időtartamának a végét, ekkor a RESET jel beáll „0”-ra. Az aktív időtartam az RC szorzattól függ (ezt az áramkör *időállandójának* is nevezik), annál hosszabb, minél nagyobb ennek értéke. A dióda szerepe az, hogy a gép

kikapcsolásánál süsse ki a feltöltött állapotban lévő kondenzátort. A jelek időbeli alakulását a következő idődiagramon lehet követni:



A RESET impulzus időtartamának (τ) van minimális értéke, ami alatt a jelnek nincs hatása. Ezt az értéket a gyártó adja meg a processzor műszaki adataiban.

TÉTELEK, KÉRDÉSEK

- 2.1. Mi az összefüggés egy memória kapacitása és a címzéséhez szükséges bitek száma között? Mondjon egy példát.
- 2.2. Rajzolja le a modellszámítógép felépítését!
- 2.3. Vázolja fel egy memória-olvasási ciklus idődiagramját és magyarázza el a lefolyását!
- 2.4. Mikor és miként kell várakozó állapotokat beiktatni az olvasási ciklusba?
- 2.5. Vázolja fel egy memória-írási ciklus idődiagramját és magyarázza el a lefolyását!
- 2.6. Ismertesse a processzor és a buszra kapcsolt többi egység áramerősségei közötti feltételt, a helyes logikai szintek biztosításának érdekében!
- 2.7. Vázolja fel és magyarázza meg a cím- és adatbusz kialakításának elvét multiplexált buszt használó processzor esetében!
- 2.8. Ismertesse a kristályoszillátor előnyét és működési elvét!

2.9. Vázolja fel egy processzor RESET áramkörének kapcsolási rajzát és ismertesse a működési elvét!

3. MODELLPROCESSZOR. UTASÍTÁSKÉSZLET TERVEZÉSE. PROGRAMOZÁS.

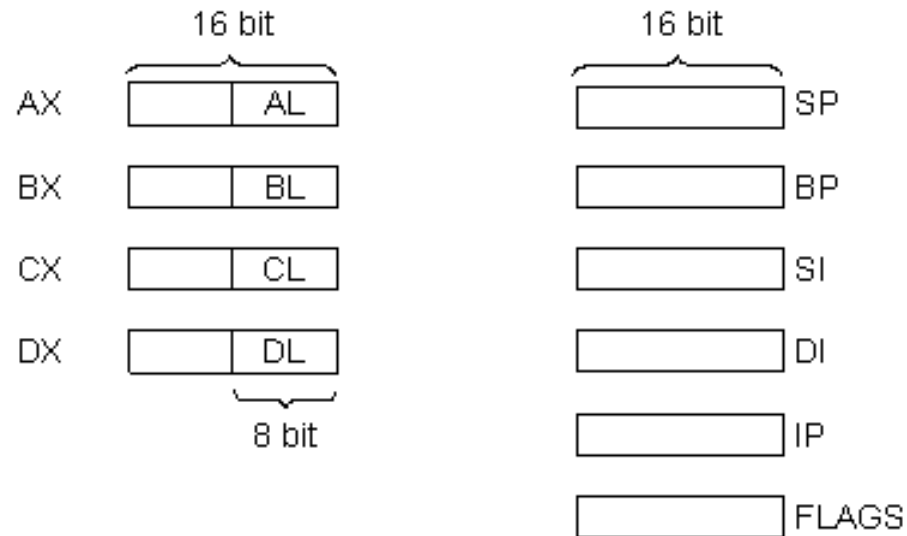
[Kovács M., Knapp G., Ágoston Gy., Budai A. – Bevezetés a számítástechnikába, LSI, 2002, VI. fejezete nyomán]

3.1. Regiszterkészlet

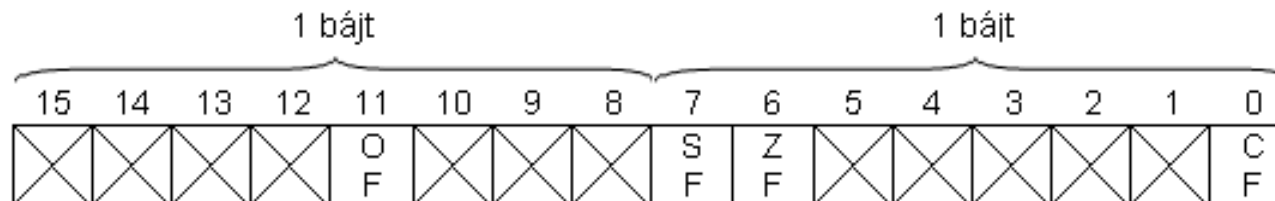
A modellprocesszorunk **regiszterkészletét** a következőképpen határozzuk meg:

AX	<i>Akkumulátor</i> (Accumulator)
BX	<i>Bázisregiszter</i> (Basis register)
CX	<i>Számláló regiszter</i> (Counter register)
DX	<i>Adatregiszter</i> (Data register)
SP	<i>Veremmutató</i> (Stack pointer)
BP	<i>Bázismutató</i> (Basis pointer)
SI	<i>S-Indexregiszter</i> (Source index)
DI	<i>D-Indexregiszter</i> (Destination index)
IP	<i>Utasítászámláló</i> (Instruction pointer)
FLAGS	<i>Állapotjelző</i>

Mindegyik regiszter 16 bites, de az első négy regiszter alsó felét (AI, BL, CL, DL) külön is lehet használni bájtműveletekre. A regiszterkészletet az alábbi ábra mutatja:



A **FLAGS állapotjelző regiszter** bitjei különösen a feltételes ugróutasítások kivitelezése szempontjából fontosak. Ez a regiszter modellünkben szintén 16 bitet fog tartalmazni a következő ábra szerint (az X-szel jelölt bitek nem használatosak):



Az ábra jelöléseivel az egyes flag-ek jelentése a következő:

- **CF (Carry flag)** Értéke 1, ha egy művelet eredményének legmagasabb helyi értékű (2^{15}) bitjén átvitel keletkezik
- **ZF (Zero flag)** Értéke 1, ha egy művelet eredményének minden bitje 0
- **SF (Sign flag)** Megegyezik a művelet eredményének legmagasabb helyi értékű bitjével (azaz a kettes komplement ábrázolás szerinti értéke 1, ha az eredmény negatív)

- **OF (Overflow flag)** Előjeles értékkel végzett műveleteknél a túlsordulást jelzi, azaz $OF = 1$, ha az eredmény nem előjelhelyes

Természetesen egy valódi processzorban a FLAGS regiszter általunk „nem használt”-nak minősített bitjei további fontos információkat tartalmaznak a processzor állapotáról, illetve vezérlik a processzor működését. Ilyen például: a paritáshiba jelzése, a hardver megszakítások letiltása stb.

3.2. Címzési módok

A processzormodellünk megalkotásában fontos szempont volt, hogy a legelterjedtebben használatos Intel processzoroktól csak a legszükségesebb mértékben térjünk el. Ezért elképzelt processzorunk jó közelítéssel egy erősen egyszerűsített, *primitív Intel processzornak* tekinthető. (Az Assembly utasítások szimbolikus nevei is azonosak az IBM PC Assemblyben megfelelő mnemonikjaival.)

A **modellprocesszor címzési módjainak** kidolgozása során arra a kérdésre kell megkeresnünk a választ, hogy a gépi utasítás-végrehajtás különböző „szereplőit”:

- az általános regisztereket,
- a memória 1, illetve 2 bájtos egységeit,
- az I/O portok regisztereit,

milyen módon tudjuk azonosítani a programutasításokban. Ezekhez a számítógép bináris működése miatt nyilvánvalóan bináris kódokat kell hozzárendelni és egy kódnak egyértelműen azonosítani kell egy tárolóegységet.

A **regisztereket** egyszerűen 4-bites *bináris sorszámukkal címezzük* a következő felsorolás szerint:

Binárisan	Decimálisan	Hexadecimálisan	Szimbolikus nevük
0000	0	0	AX
0001	1	1	AL
0010	2	2	BX
0011	3	3	BL
0100	4	4	CX
0101	5	5	CL
0110	6	6	DX
0111	7	7	DL
1000	8	8	SP
1001	9	9	BP
1010	10	A	SI
1011	11	B	DI
1100	12	C	IP

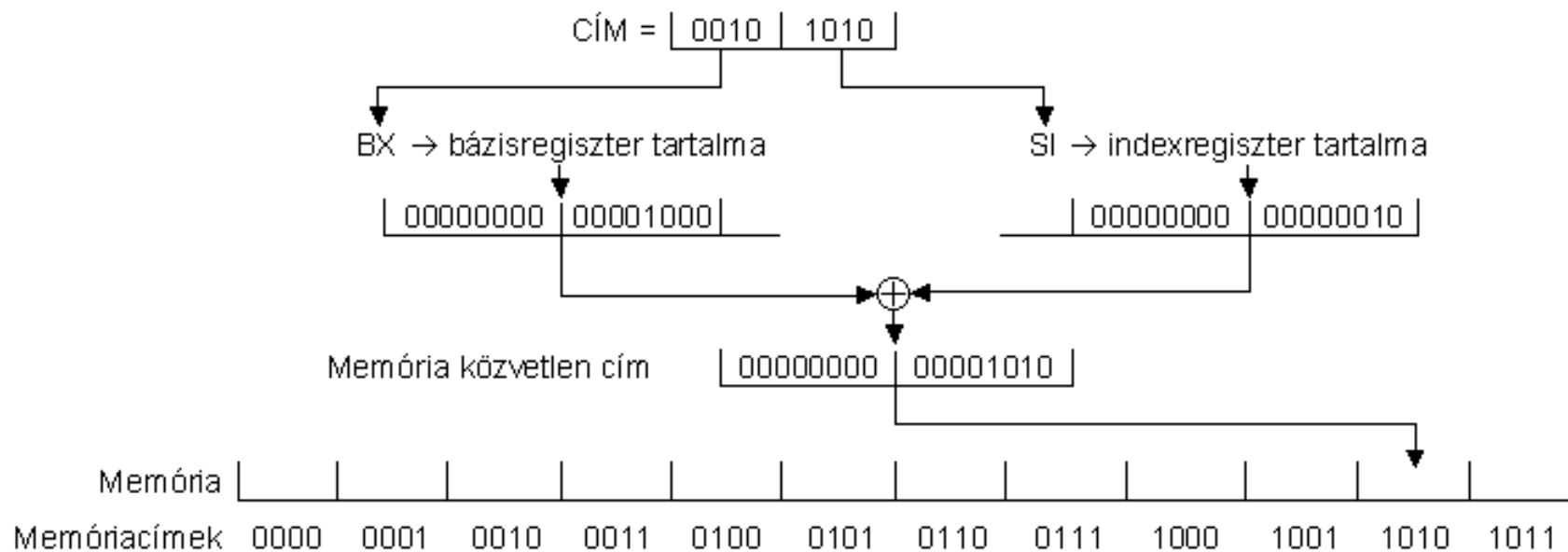
3.1. táblázat. A regiszterek bináris kódjai

Felvetődhet a kérdés, hogy a FLAGS regisztert miért nem láttuk el egy bináris azonosítóköddel. Ennek az az oka, hogy ezt a regisztert közvetlenül a gépi utasításokban nem lehet megcímezni. Ugyanakkor külön utasításokkal a FLAGS regiszter tartalmához is hozzáférhetünk.

A memória címzéséhez háromféle módot engedünk meg processzor modellünkben. Ezek:

- a regiszter indirekt címzése, ekkor az utasításban szereplő regiszter tartalmazza azt a 16-bites memóriacímet, ahol az adat megtalálható;
- a közvetlen adatkímzés, amikor magában az utasításban található meg egy 8-, illetve 16-bites adat- vagy címérték;
- a bázis-relatív és indexelt címzés, amikor az utasítás két regiszter címét tartalmazza.

Ebben az esetben az adat tényleges 16-bites memóriacímét úgy számítjuk ki, hogy a két regiszterben található címet összeadjuk (lásd az ábrát).



Végezetül az **I/O portok címzését** határozzuk meg, melynél figyelembe kell venni, hogy a portban található két regisztert önállóan meg kell címeznünk az utasításokban. Ezért e portok regisztereinek a címeit a következőképpen definiáljuk:

- 0 port puffer regiszter 00
- 0 port állapotregiszter 01
- 1 port puffer regiszter 10
- 1 port állapotregiszter 11

Foglaljuk össze, hogy a programutasításokban szereplő tárolóelemeket milyen kódokkal azonosítjuk:

- | | |
|--|----------------------|
| a/ Általános regiszterek | 4-bites regiszterkód |
| b/ Memória indirekt címzése regiszterrel | 4-bites regiszterkód |

c/ Közvetlen adatszám	8- és 16-bites adatszám
d/ Memória bázis-relatív és -indexelt címezése	két 4-bites regiszter kód
e/ I/O portok regisztereinek címezése	2-bites port kód

Vegyük észre a következő problémákat:

- Ha az utasításban egy operandus helyén egy regiszter kódja van, hogyan ismeri fel a processzor, hogy a regiszter egy adatot vagy egy címet tartalmaz (Másként fogalmazva, mi alapján lehet elkülöníteni az a/ és b/ esetet?);
- Ha az utasítás egy 8 bites bináris számot tartalmaz, mi alapján dönthető el, hogy ez egy közvetlen adatszámhoz tartozó bájt vagy pedig két regiszter azonosítója (c/ és d/ eset).

A többértelmezőség kiküszöbölése érdekében az a/, b/, c/, d/ eseteknek megfelelő **címezési módokat binárisan fogjuk kódolni** a következőképpen:

- Regiszterben lévő adat 00 = R
- Regiszter indirekt címezése 01 = I
- Közvetlen adatszám 10 = D
- Bázis-relatív és -indexelt címezés 11 = B

Az elkövetkezendőkben az a/, b/, c/, d/ címezési módokat rendre R, I, D, B-vel fogjuk jelölni. Az I/O port regisztereinek a címezését külön egyedi esetként kezeljük, mivel ezt az I/O utasítás kódja egyértelműen azonosítja.

3.3. A gépi utasítás szerkezete

Mint később látni fogjuk, modellprocesszorunk egy gépi utasítása bináris formában és Assembly mnemonikkal a következő ábra szerinti formátumú:

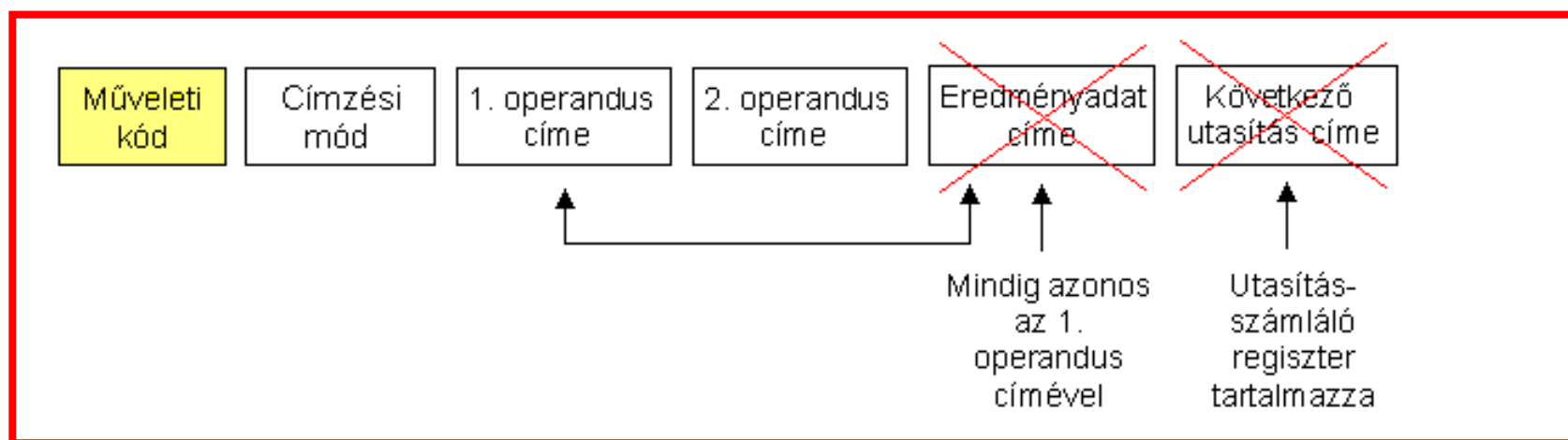
Assembly formátum:	ADD AX, [BX+SI]
Gépi kód:	0000 0000 0010 1010

Ez egy bázisindexelt címzésű két bájtos utasítás. Figyeljük meg az ábrán látható bitsorozatot és tegyük fel a kérdést: honnan állapítja meg a processzor, hogy a bitek melyik csoportja jelenti a műveleti kódot (mit kell elvégezni) és az utasításban résztvevő tárolók címét?

Nyilvánvalóan, ehhez egy egyértelmű és konzekvensen alkalmazott szabályrendszer kell, mely meghatározza, hogy az utasításokban lévő biteket a processzor hogyan értelmezze. Ezt határozza meg a **gépi utasítás szerkezete**, melyet most fogunk kidolgozni modellprocesszorunk számára.

Egy gépi utasítás funkcionális részekre való felbontását az határozza meg, hogy a processzornak milyen információra van szüksége az utasítás végrehajtásához.

Ha egy gépi utasításban maximum két adat vehet részt, akkor a művelet végrehajtásához a processzornak ismernie kell a művelet kódját, az 1. és 2. operandus címét, annak a tárolóhelynek a címét ahol az eredményt el kell helyezni, valamint azt a címet, ahol a programot folytatni kell, ha az utasítás végrehajtása befejeződött:



A következő utasítás címe és az eredményadat címe részeket kihagyhatjuk a gépi utasításból a következő két konvenció bevezetésével:

- a következő utasítás címét mindig az IP utasításszámláló regiszter tartalmazza;

- két operandust tartalmazó gépi utasításoknál az eredményadatot mindig az első operandus címén (annak korábbi állapotát felülírva) kell elhelyezni.

Ezt figyelembe véve modellszámítógépünk utasításkészletében maximum kétcímű utasításokat fogunk szerepeltetni.

Az utasításszerkezet specifikálása során az lesz a célunk, hogy a megtervezett utasítások minél rövidebbek legyenek, tehát ahol ez lehetséges takarékoskodni fogunk a bitekkel. Ennek az az értelme, hogy a rövidebb gépi utasításokból álló program egyrészt kevesebb tárolóhelyet igényel, másrészt végrehajtása is gyorsabb, mivel az utasítások kiolvasása során kisebb mennyiségű adatot kell mozgatni a főtár és a processzor között.

Az előbbieket figyelembe véve modellprocesszorunk gépi utasítása négy, három illetve két főrészből fog állni:



Próbálkozzunk most azzal, hogy a műveleti kód és a címzési mód mezőkre ne használjunk el többet 1 bájt nál. Mivel az R, I, D, B címzési módok 2 biten kódolhatóak, ezért a **címzési mód mező**

- kétcímű utasításnál 4 bit,
- egy című utasításnál 2 bit,
- 0 című utasításnál 0 bit

tárolóhelyet igényel. Ebben az esetben a műveleti kód kódolására

- kétcímű utasításnál 4 bit,
- egycímű utasításnál 6 bit,
- 0 című utasításnál 8 bit

lehetőségünk marad.

A műveleti kódot OP-val, a címezési módot AM jelölve ezt mutatja be a következő ábra:

	Bitpozíció	7	6	5	4	3	2	1	0
Kétcímű utasítások		OP				AM			
Egycímű utasítások		OP						AM	
0 című utasítások		OP							

Ennek megfelelően az utasításkészletben például maximum 16 (2^4) kétcímű utasítást leszünk képesek kódolni. Ha ezekbe a keretekbe az utasításkészletbe kiválasztott konkrét utasítások beleférnek, akkor a műveleti kód és a címezési mód bináris kódolását 1 bajton meg tudjuk oldani.

Tekintsük most át utasításaink **címrészének felépítését**. Ehhez emlékeztetünk arra, hogy a címezési mód szerint:

- R 4 bit
- I 4 bit
- D 16 bit (8 bit)
- B $4 + 4 = 8$ bit

tárolóhelyre van szükségünk.

Itt beleütköztünk abba a problémába, hogy 8- és 16-bites adataegységeket is szeretnénk mozgatni a főtár és a CPU között. Hogyan lehet ezt kifejezni az utasításszerkezetben, mivel a kódolásra felhasználható bitjeink már „elfogytak”.

Ezt az ellentmondást a következő **konvencióval** küszöbölhetjük ki: *ha az 1-es operandusban 16- illetve 8-bites regiszter szerepel, akkor ennek megfelelően a 2-es operandus által meghatározott memóriacímről 2 illetve 1 bájtot kell átvinni a CPU-ba.*

Az utasítások címrészének felépítése szerint elvileg a következő **utasítástípusok** lehetségesek:

- *Kétcímű utasítások:* RR, RI, RD, RB,
IR, II, ID, IB,
BR, BI, BD, BB
- *Egycímű utasítások:* R, I, D, B
- *0 című utasítások*

Ehhez két megjegyzést célszerű tenni:

- Az utasítástípusok most még csak „elméletileg” léteznek, ezeket a variációkat majd a konkrét utasítások definiálásakor töltjük fel tartalommal.
- A DR, DI stb. utasítástípusok nyilvánvalóan nem értelmesek, mert konvenciónk szerint az utasítás eredménye az 1-es operandusban keletkezik.

A modellprocesszorunk utasításszerkezetének teljeskörű definícióját az alábbi ábra tartalmazza.

Utasítás típus	1 bájt		1 bájt		1 bájt		1 bájt		
RR	OP	00 00	R1	R2					K É T C Í M Ű
RI	OP	00 01	R1	R2					
RD	OP	00 10	R1	0000	DB				
				0000	DW				
RB	OP	00 11	R1	0000	R2	R3			
IR	OP	01 00	R1	R2					
II	OP	01 01	R1	R2					
ID	OP	01 10	R1	0000	DB				
				0000	DW				
IB	OP	01 11	R1	0000	R2	R3			
BR	OP	11 00	R1	R2	0000	R3			
BI	OP	11 01	R1	R2	0000	R3			
BD	OP	11 10	R1	R2	DB				
					DW				
BB	OP	11 11	R1	R2	R3	R4			E G Y C Í M Ű
R	OP	00	0000	R1					
I	OP	01	0000	R1					
D	OP	10	DB						
			DW						
B	OP	11	R1	R2					
0 című utasítás	OP								0 című

Jelölések:

	= Műveleti kód	R1, R2, R3, R4	= Az utasításokban szereplő regiszterek
	= 1-es operandus	DB	= 1 bájtnyi adat
	= 2-es operandus	DW	= 2 bájtnyi adat

3.2. táblázat. A modellprocesszor utasításszerkezetének definíciója

Az ábra alapján látható, hogy

- a kétcímű utasítások 2, 3 és 4 bájtt,
- az egycíműek 2, 3 bájtt
- a 0 címűek 1 bájtt

hosszúak.

3.4. Az utasításkészlethez tartozó utasítások kiválasztása és kódolása

Az elkövetkezendőkben sorra vesszük a számítógép különböző utasításcsoportjait és a funkciókat értékelve meghatározzuk, hogy modellprocesszorunk utasításkészletében milyen utasításokat szeretnénk szerepeltetni. A kiválasztásnál arra törekszünk, hogy a leggyakrabban használt Assembly utasítások szerepeljenek az utasításkészletben és ezekből olyan választék legyen, mely programok megírását is lehetővé teszi.

Aritmetikai és logikai utasítások

Az **aritmetikai műveletek** közül csak a legszükségesebbeket választjuk ki. Ezek:

- ADD összeadás (fixpontos)
- SUB kivonás (fixpontos)
- CMP összehasonlítás
- INC inkrementálás, azaz 1-gyel növelés
- DEC dekrementálás, azaz 1-gyel csökkentés

- NEG *komplementálás, azaz (-1)-gyel való szorzás (kettes komplement).*

A **logikai műveletek** közül négyet szerepeltetünk modellprocesszorunk utasításkészletében. Ezek:

- AND *logikai „ÉS”*
- OR *logikai „VAGY”*
- XOR *logikai kizáró „VAGY”*
- NOT *logikai „NEM” (egyes komplement)*

Adatmozgató utasítások

Az adatmozgató utasításokat úgy fogjuk kiválasztani, hogy segítségükkel

- a memória és regiszter
- a regiszter és regiszter
- a regiszter és verem
- a regiszter és I/O portok

közötti adatátvitel végrehajtható legyen. Ezek:

- MOV *adatmozgatás memória és regiszter, regiszter és regiszter között*
- PUSH *regiszter betöltése verembe*
- POP *verem tetejének betöltése regiszterbe*
- PUSHF *FLAGS regiszter mentése verembe*
- POPF *FLAGS regiszter betöltése veremből*
- IN *adatolvasás I/O portról*
- OUT *adatküldés I/O portra.*

Vezérlésátadó utasítások

E körbe tartozó utasítások közül a ciklusképzéshez, a szubrutinokhoz, a feltételes és feltétel nélküli ugráshoz, valamint a programvezérelt megszakításhoz szükséges műveleteket választjuk ki. Ezek:

- LOOP *CX értékének dekrementálása és ugrás, ha $CX \neq 0$*
- CALL *ugrás szubrutinba (szubrutin felhívása)*
- RET *visszatérés szubrutinból*
- JMP *feltétel nélküli ugrás az utasításban megadott címre*
- JZ/JE *ugrás, ha $ZF = 1$ / ugrás, ha egyenlő (CMP után)*
- JNZ/JNE *ugrás ha $ZF = 0$ / ugrás, ha nem egyenlő (CMP után)*
- JC/JB *ugrás, ha $CF = 1$ / ugrás, ha kisebb*
- JNC/JAE *ugrás ha $CF = 0$ / ugrás, ha nagyobb egyenlő*
- INT *programvezérelt megszakításkérés*
- IRET *visszatérés megszakítás-kiszolgáló rutinból.*

Bitléptető és forgató utasítások

- SHL *Logikai eltolás balra*
- SHR *Logikai eltolás jobbra*
- RCL *Forgatás balra CF-fel*
- RCR *Forgatás jobbra CF-fel*

A bitléptető és forgató utasításokat a fejezet második részében bemutatott mintaprogramokban fogjuk felhasználni. Néhány ilyen utasítást azért szerepeltettünk az utasításkészletben, mert ezek nélkül utasításkészletünk nem reprezentálná a valódi processzorokat. Érdeemes felhívni a figyelmet arra, hogy bár a szorzás és osztás művelete nincs benne modellparancsunk utasításkészletében, ezek is elvégezhetők ismételt összeadással és a léptető utasítások felhasználásával.

Processzorvezérlő utasítások

- CLC *CF beállítása 0-ra*
- STC *CF beállítása 1-re*
- HLT *Processzor leállítása egy hardver megszakításig*

Miután funkcionálisan rögzítettük modellprocesszorunk utasításkészletét, a következőkben meghatározzuk az **egyes utasítások műveleti kódját**. Ehhez az egyes utasításokban elvégzendő műveletek értékelésével először meghatározzuk, hogy utasításaink közül melyek a kétcímesek, egycímesek illetve 0-címűek. (A processzorvezérlő utasítások, a PUSHF és POPF, valamint a RET és IRET 0-című utasítások).

Soroljuk fel először a kétcímű, majd az egycíműeket, végül a 0 című utasításokat és mindegyikhez rendeljünk hozzá rendre a 4, 6, 8 bites műveleti kódokat (lásd az ábrát).

Utasításkód mnemonik	Műveleti kód binárisan			
	4 bit	2 bit	2 bit	
ADD	0000			KÉT- CÍMŰ UTA- SÍ- TÁ- SOK
SUB	0001			
CMP	0010			
AND	0011			
OR	0100			
XOR	0101			
MOV	0110			
SHL	0111			
SHR	1000			
RCL	1001			
RCR	1010			
INC	1011	00		EGY- CÍMŰ UTA- SÍ- TÁ- SOK
DEC	1011	01		
NEG	1011	10		
NOT	1011	11		
PUSH	1100	00		
POP	1100	01		
IN	1100	10		
OUT	1100	11		
LOOP	1101	00		
CALL	1101	01		
JMP	1101	10		
JZ/JE	1101	11		
JNZ/JNE	1110	00		
JC/JB	1110	01		
JNC/JAE	1110	10		
INT	1110	11		
RET	1111	00	00	0 CÍ- MŰ UTA- SÍ- TÁ- SOK
IRET	1111	00	01	
PUSHF	1111	00	10	
POPF	1111	00	11	
CLC	1111	01	00	
STC	1111	01	01	
HLT	1111	01	10	

3.3. táblázat. A modellprocesszor utasításkódjai

Vegyük észre, hogy a műveleti kód alapján a processzor egyértelműen el tudja dönteni, hogy kétcímű, egycímű vagy 0 című utasítást olvasott be:

műveleti kód első 4 bitje ≤ 1010	$\rightarrow$	kétcímű utasítás
$1010 < \text{műveleti kód első 4 bitje} \leq 1110$	$\rightarrow$	egycímű utasítás
műveleti kód első 4 bitje $= 1111$	$\rightarrow$	0 című utasítás

3.5. Utasításkészlet gépi kódban. Programkészítés.

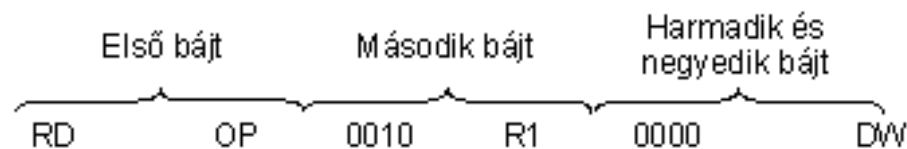
Tulajdonképpen már minden információval rendelkezünk ahhoz, hogy a gépi kódú utasításokat binárisan le tudjuk írni. Nézzünk erre példákat! Az egyes, konkrét utasítások meghatározásához semmi mást nem kell tennünk, csak a 3.1, 3.2, 3.3 táblázat adatait felhasználnunk.

1. Példa

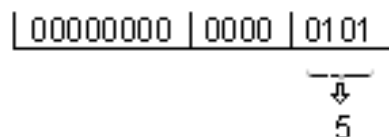
Adjuk meg annak az utasításnak a gépi kódját, mely az AX akkumulátorban lévő értéket 5-tel megnöveli.

Megoldás

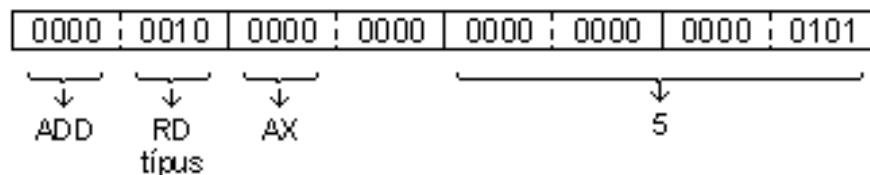
- A szükséges utasítás nyilvánvalóan az összeadás lesz.
- AX-hez egy konkrét számot kell hozzáadnunk, így az 1-es operandus egy regiszter, a 2-es operandus pedig a közvetlen adatszámítású „5”-ös szám. Az utasítástípus tehát RD.
- A 3.1. számú táblázat szerint AX kódja: 0000.
- A 3.3. számú táblázatból kikeressük az összeadás (ADD) műveleti kódját, ez: 0000
- A 3.2. számú ábra táblázatból kikeressük az RD utasítástípust. A megfelelő sor ezt tartalmazza:



Itt OP az ADD műveleti kódja, tehát 0000
 R1 az AX-t jelenti, tehát 000
 DW az 5-ös számot jelenti 16 biten binárisan kódolva.
 Ennek értéke tehát:



Behelyettesítve RD utasításszerkezetébe a keresett utasítás gépi kódját a következő ábra mutatja:



2. Példa

Töltsük fel bináris 0-val a CX regisztert! (Ezt a szakmai zsargonban legtöbbször úgy fejezik ki, hogy töröljük CX-et)

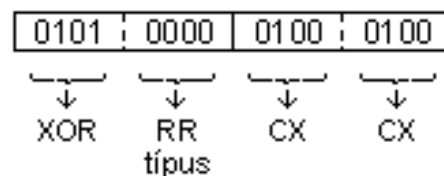
Megoldás

Ezt a feladatot az 1. példához hasonlóan a MOV utasítással is megoldhatjuk (ezt gyakorlásként ajánljuk az olvasónak), de „elegánsabb” és gyorsabb megoldás az XOR „kizáró vagy” utasítás alkalmazása. Az XOR CX, CX utasítás végrehajtása után ugyanis CX tartalma bináris 0 lesz.

Az XOR műveleti kódja: 0101

A CX regiszter kódja: 0100

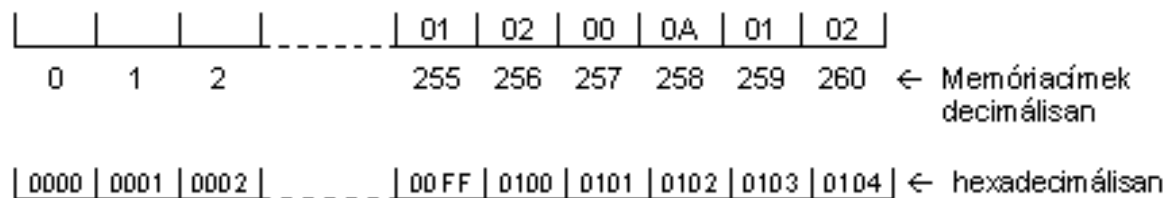
Utasításunk nyilván RR típusú, ezért a keresett gépi kód a következő:



3. Példa

Írjunk egy kis programrészt a következő feladatra:

A memória 255-ik bájtyától kezdődően 6 db 1 bájtos szám található, amint az alábbi ábra mutatja:



Adjuk össze ezt a 6 db számot úgy, hogy az eredmény AX-ben keletkezzen.

1. Megoldás

```
XOR  AX, AX    ; Először AX-et és DX-et törölni kell,  
XOR  DX, DX    ; mert nem tudjuk mi a tartalma  
MOV  BX, 255   ; BX-be az első bájttípusának elhelyezése  
MOV  DL, [BX]  ; Vigyél DX alsó 1 bájtyába a  
               ; BX regiszterben található címről 1 bájtot  
ADD  AX, DX    ; Add össze AX-et és DX-et (az eredmény  
               ; AX-ben keletkezik)  
MOV  BX, 256   ; Az előző 3 utasítást most még 5-ször  
MOV  DL, [BX]  ; ismételni kell a fennmaradó 5 bájtt
```

```

ADD  AX, DX      ; összeadásához
MOV  BX, 257
MOV  DL, [BX]
ADD  AX, DX
MOV  BX, 258
MOV  DL, [BX]
ADD  AX, DX
MOV  BX, 259
MOV  DL, [BX]
ADD  AX, DX
MOV  BX, 260
MOV  DL, [BX]
ADD  AX, DX

```

Tegyük fel most a kérdést: hogyan oldanánk meg ezt a feladatot, ha nem 6 db, hanem mondjuk 600 számot kellene összeadni? Nyilván nem szeretnénk a két MOV és az egy ADD utasítást 600-szor megismételni.

2. Megoldás

Vegyük észre, hogy az előző programrészletben a

```

MOV BX, ?
MOV DL, [BX]
ADD AX, DX

```

utasításokat úgy ismételtük, hogy csak a ? -el jelölt értéket (címet) változtattuk. Ez a cím először 255 volt, azután 256, 257 stb., azaz mindig 1-gyel növekedett. BX értékének 1-gyel való megnövelését viszont megoldhatjuk az INC utasítással.

Ha BX értéke már 255, akkor a következő két utasítássorozat teljesen egyenértékű:

$$\left. \begin{array}{l} \text{MOV BX, 256} \\ \text{MOV DL, [BX]} \\ \text{ADD AX, DX} \end{array} \right\} \equiv \left\{ \begin{array}{l} \text{INC BX} \\ \text{MOV DL, [BX]} \\ \text{ADD AX, DX} \end{array} \right.$$

A jobboldali utasítássorozat már nem tartalmazza a 256, 257 stb. konstansokat. Ezért, ha azt mondhatnánk, hogy ezt a három utasítást ismételd meg hatszor, akkor célhoz jutnánk.

Ezt úgy fogjuk elérni, hogy a CX számlálóregiszter kezdeti értékét 6-tal (a számok darabszámával) feltöltjük, majd minden egyes szám hozzáadását követően CX értékét 1-gyel csökkentjük (DEC utasítás).

Végül egy JNZ (ugrás ha ZF = 0, azaz CX ≠ 0) utasítással „visszaugrunk” az összeadást addig ismételni, amíg CX értéke 0 nem lesz. Ekkor már ZF = 1-re lesz állítva, tehát nem fog bekövetkezni visszaugrás.

Ezt figyelembe véve a feladatot megoldó programrészletünk a következő lesz:

1	XOR	AX, AX	
2	XOR	DX, DX	
3	MOV	BX, 255	
4	MOV	CX, 6	
5 ISMET:	MOV	DL, [BX]	
6	ADD	AX, DX	
7	INC	BX	; BX értékének növelése 1-gyel, hogy a ; következő bájt címezze meg
8	DEC	CX	; CX értékének csökkentése 1-gyel
9	JNZ	ISMET	; ugrás, ha CS ≠ 0

E programrészletben az „ISMET:” az **Assembly programnyelvben alkalmazható ún. szimbolikus címke**, melyet a fordítóprogram a MOV DL, [BX] utasítás memóriabeli kezdőcímének fog értelmezni.

A következő táblázatban feltüntetjük, hogy az előző programrészlet végrehajtása során az utasítások befejezésekor az AX, BX, CX, DX regiszterek tartalma milyen hexadecimális értéket vesz fel. (A 6 db 1 bájtos szám a 3. példa elején megadott ábra szerinti értékeket tartalmazza.)

Javasoljuk az olvasónak, hogy egy ilyen üres táblázatot először a programutasítások alapján próbáljon meg kitölteni, majd az eredményt a 3.4. számú táblázat alapján ellenőrizze.

		Utasítás sorszáma	AX	BX	CX	DX
1. ÁT FUTÁ S		1	0000	?	?	?
		2	0000	?	?	0000
		3	0000	00FF	?	0000
		4	0000	00FF	0006	0000
		5	0000	00FF	0006	0001
		6	0001	00FF	0006	0001
		7	0001	0100	0006	0001
		8	0001	0100	0005	0001
		9	0001	0100	0005	0001
2. ÁT FUTÁ S		5	0001	0100	0005	0002
		6	0003	0100	0005	0002
		7	0003	0101	0005	0002
		8	0003	0101	0004	0002
		9	0003	0101	0004	0002
3. ÁT FUTÁ S		5	0003	0101	0004	0000
		6	0003	0101	0004	0000
		7	0003	0102	0004	0000
		8	0003	0102	0003	0000
		9	0003	0102	0003	0000
4. ÁT FUTÁ S		5	0003	0102	0003	000A
		6	000D	0102	0003	000A
		7	000D	0103	0003	000A
		8	000D	0103	0002	000A
		9	000D	0103	0002	000A
5. ÁT FUTÁ S		5	000D	0103	0002	0001
		6	000E	0103	0002	0001
		7	000E	0104	0002	0001
		8	000E	0104	0001	0001
		9	000E	0104	0001	0001
6. ÁT FUTÁ S		5	000E	0104	0001	0002
		6	0010	0104	0001	0002
		7	0010	0105	0001	0002
		8	0010	0105	0000	0002
		9	0010	0105	0000	0002

Jelölés
=

A megfelelő utasítás végrehajtása során az eredményadatok helye

3.4. táblázat. A regisztertartalmak változása programrészletünk végrehajtása során

Látható, hogy programrészletünk végrehajtása során az 5, 6, 7, 8, 9 sz. utasítások 6-szor kerülnek végrehajtásra.

Ha egy utasítássorozatot egymás után többször, változatlan formában, de általában változó adatokkal hajtunk végre a program futtatása során, akkor ezt **ciklusnak** nevezzük.

Az egyik gyakori programhiba az úgynevezett „örök ciklus”. Az előző programpéldánkban ez az eset állt volna elő, ha a 4 sorszámú MOV CX, 6 utasítást az utasítássorból „kifelejtjük” és a CX regiszter tartalma a ciklusba való belépéskor 0000 vagy egy negatív szám. Ebben az esetben CX értéke a dekrementálások hatására negatív lesz és ezáltal a ZF flag sohasem fogja felvenni az 1 értéket. Emiatt a program végrehajtásakor az 5, 6, 7, 8, 9 utasításokat a számítógép „végtelen sokszor” megismétli.

Az előző programrészletet felhasználjuk arra, hogy a gépi kódú programokat is bemutassuk, és kis betekintést nyerjünk az Assembly nyelvű programok fordítóprogramjának működésébe.

Tételezzük fel, hogy programrészletünk a memóriában közvetlenül a 6 db 1 bájtos szám után helyezkedik el. Ekkor kezdőcíme decimálisan 261, hexadecimálisan pedig 0105. A korábban már megismert módon állítsuk elő a 9 db utasítás gépi kódját:

	1 bájt		1 bájt		1 bájt		1 bájt	
XOR AX, AX	0101	0000	0000	0000				
XOR DX, DX	0101	0000	0110	0110				
MOV BX, 255	0110	0010	0010	0000	0000	0000	1111	1111
MOV CX, 6	0110	0010	0100	0000	0000	0000	0000	0110
ISMET: MOV DL, [BX]	0110	0001	0111	0010				
ADD AX, DX	0000	0000	0000	0110				
INC BX	1011	0000	0000	0010				
DEC CX	1011	0100	0000	0100				
JNZ ISMET	1110	0010	0000	0001	0001	0001		

3.5. táblázat. Programrészletünk gépi kódban

A 3.5. számú táblázatot a 3.1., 3.2. és 3.3. számú táblázatok alapján állítottuk össze. Ennek során a lényegét tekintve a **fordítóprogramot szimuláltuk**:

- A szimbolikus műveleti kód mnemonikok (pl. XOR, MOV stb.) alapján kikerestük a műveleti kód táblázatból a műveleti kód bináris értékét.
- „Felismertük” a regiszterek szimbolikus nevét (AX, BX, stb.) ez alapján a regiszterkódtáblázatból kikerestük a regiszterek bináris kódját.
- Az utasításokban található regiszter darabszám, konstansok, az indirekt címezést jelölő [] jel alapján meghatároztuk, hogy az utasításoknak milyen az utasításszerkezete. Ezt követően az utasításszerkezetre vonatkozó táblázatból kikerestük az utasítás formai felépítését.
- Ha mindezt megtettük, ezt követően már csak az utasításokban lévő konstansokat kellett binárisra átkonvertálni és a szimbolikus címeknek (ISMET:) a bináris értékét meghatározni. Ezt követően kitölthettük a 3.5. számú táblázatot gépi kódban.

A **szimbolikus címek bináris értékének meghatározása** még további magyarázatot igényel.

Mint már említettük az Assembly konvenciók szerint az „ISMET:” szimbolikus cím a MOV DL, [BX] utasítás gépi kódjának tárolóbeli címét jelenti. Ahhoz, hogy a JNZ utasításban lévő ugrási cím (mely szimbolikusan ISMET:) bináris értékét meghatározzuk egy rövid számításra van szükség.

Ennek során először az ISMET: címkével jelölt MOV utasítás gépi kódjának a programrészlet elejétől való távolságát határozzuk meg bájtokban számolva.

A 3.5. számú táblázat szerint az

ISMET: MOV DL, [BX]

utasítást 2 darab 2 bájtos (2 db. XOR) és 2 darab 4 bájtos (2 db. MOV) utasítás előzi meg. Ez összesen

$$2 \times 2 + 2 \times 4 = 12$$

bájtot jelent. Ez azt jelenti, hogy az ISMET: szimbolikus címnek megfelelő cím a programrészlet kezdőcímétől 12 bájtra található. Mivel tudjuk, hogy programrészletünk kezdő címe a memóriában 261, ezért ISMET: címe:

$$261 + 12 = 273$$

vagy hexadecimálisan és binárisan:

0111H = 0000 0001 0001 0001

Példánk alapján világos, hogy az Assembly nyelvű programokból egy jól felépített szabály és eljárás rendszer alapján előállíthatjuk az utasítások gépi kódját. Erre az algoritmusra tehát programot is írhatunk, amit **Assembler**nek nevezünk. Az Assembler tehát az a *fordítóprogram, mely az Assembly nyelven megadott programutasításokból gépi kódú utasításokat állít elő.*

TÉTELEK, KÉRDÉSEK

- 3.1. Mi a szerepe a FLAGS regiszternek, és milyen biteket tartalmaz?
- 3.2. Mit kell megadni az utasítás címrészeiben, és hol található az operandus a következő címezési módoknál:
 - regiszterben lévő adat
 - regiszter indirekt címezése
 - közvetlen adatszímzés
 - bázis-relatív és -indexelt címezés
- 3.3. Váolja a két operandust feldolgozó utasítás logikai szerkezetét! Mely mezőket lehet ebből kihagyni, és miért?
- 3.4. Adjon öt példát aritmetikai és logikai utasításokra!
- 3.5. Adjon öt példát adatmozgató utasításokra!
- 3.6. Adjon öt példát vezérlésátadó utasításokra!
- 3.7. Mi a szimbolikus címke, és hogy kezeli a fordítóprogram?

4. HUZALOZOTT ÉS MIKROPROGRAMOZOTT VEZÉRLŐEGYSÉG TERVEZÉSI KÉRDÉSEI

4.1. A vezérlőegység feladata

Egy utasítás végrehajtása több ciklus végrehajtását jelenti, mint az utasítás lehívása, oparandus címszámítása, végrehajtás, megszakítás. Az utasításciklus minden ilyen kisebb alciklusa (fázisa) elemi lépések sorozatát feltételezi, amelyekben olyan egyszerű műveletek zajlanak, mint két regiszter közötti adatátvitel, egy regiszter és a külső busz közötti átvitel, vagy egyszerűen egy aritmetikai művelet. Megállapítható tehát, hogy *egy utasításciklus minden fázisát **elemi műveletek** sorozatára lehet lebontani, amelyeket egy-egy időegység alatt kell végrehajtani:*

PROGRAM VÉGREHAJTÁSA→UTASÍTÁSCIKLUSOK→ALCIKLUSOK (lehívás, címszámítás, stb.)→ELEMI MŰVELETEK

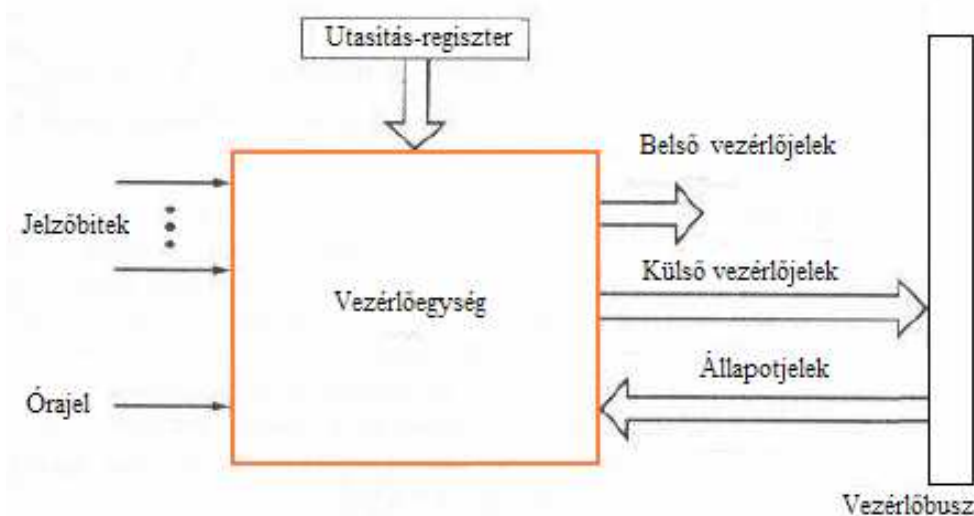
A processzor **vezérlőegysége** (*control unit*) felel minden utasítás elemi műveletének helyes sorrendben való végrehajtásáért, aminek érdekében a megfelelő időpillanatokban vezérlőjeleket bocsát ki. A vezérlőegység jellemzése több szempontot feltételez:

1. A processzor *alapegységeinek* a meghatározása (ALU, regiszterek, belső buszok, külső buszok, vezérlőegység).
2. A processzor által végrehajtandó *utasítások* specifikálása.
3. Az utasítások által feltételezett *elemi műveletek* meghatározása.
4. A vezérlőegységnek az elemi műveletek végrehajtására irányuló *feladatainak* a meghatározása.

A vezérlőegységnek két alapfeladata van:

- Az elemi lépések *megfelelő sorrendben* való végrehajtása, a feldolgozott programnak megfelelően;
- Az elemi műveletek végrehajtásához szükséges *vezérlőjelek* generálása.

A vezérlőegységnek egy általános modelljét mutatja a következő ábra:



A vezérlőegység bemenetei:

- **Órajel** – minden órajelütemben egy-egy elemi művelet vagy egyidejű műveletcsoport kerül végrehajtásra;
- **Utasításregiszter** – az aktuális utasítás műveleti kódja határozza meg, hogy milyen elemi lépéseket kell elvégezni a végrehajtási ciklusban;
- **Állapotjelzők** – megadják a processzor állapotát (hol tart az utasításcikluson belül) és az előző ALU-művelet kimenetét;
- **Vezérlőjelek a vezérlőbuszról** – külső egységek megszakításait és visszajelzéseit tartalmazzák.

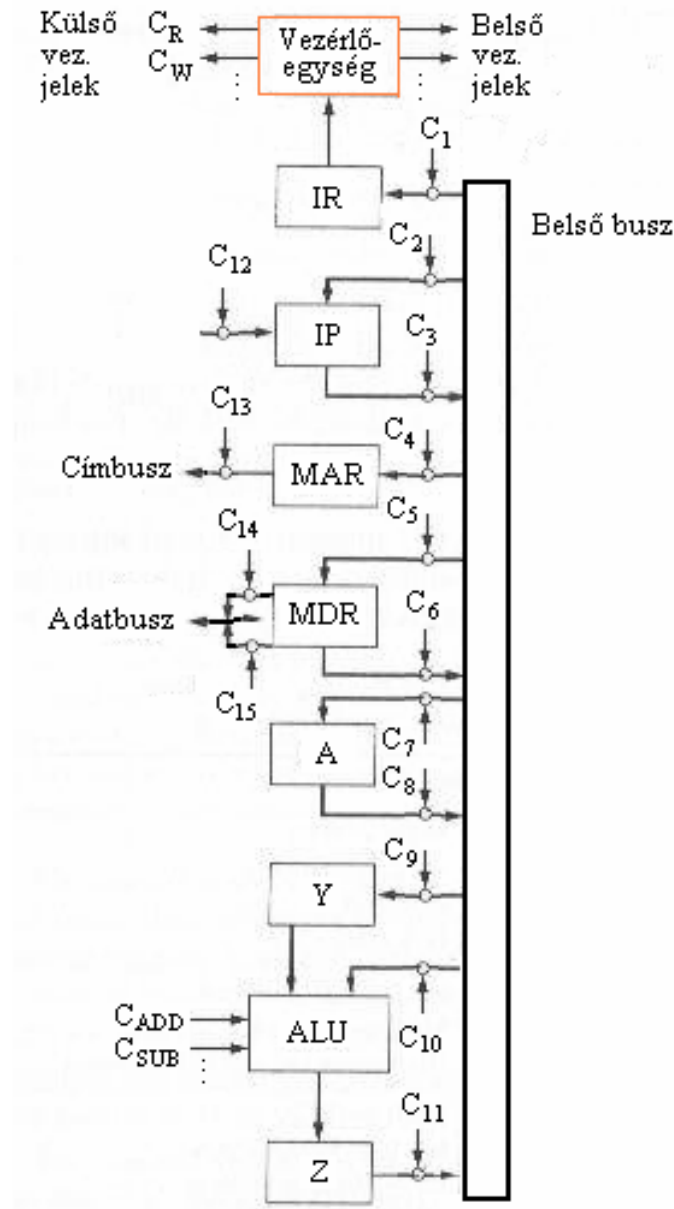
A vezérlőegység kimenetei:

- **Belső vezérlőjelek** – a regiszterek közötti adatátvitelt vezérik, illetve az ALU műveletét határozzák meg;
- **Külső vezérlőjelek** – a memóriát és az I/O portokat vezérik.

Minden órajelütemben a vezérlőegység beolvassa a bemeneteket, és kiadja a megfelelő elemi művelet végrehajtását biztosító **vezérlőjeleket**. A vezérlőjelek három típusát különböztetjük meg:

- az ALU egyik műveletét aktiválja;
- egy belső adatútvonalat engedélyez;
- a rendszerbuszon küld ki egy vezérlést.

Végül soron mindegyik vezérlőjel egy-egy logikai kapu bemenetére kerül, amelynek a megnyitásával történik meg egy adat regiszterbe való beírása vagy onnan való kiolvasása, valamint egy ALU-művelet kiváltása. Az alábbi ábra egy belső busz köré szervezett processzor felépítését mutatja a vezérlési pontokkal együtt:



Az ALU mellett két – eddig nem tárgyalt – Y-nal és Z-vel jelölt regiszter jelenik meg. Egy kétoperandusú művelet esetében az Y regiszter tárolja ideiglenesen az egyik operandust (a másik a belső adatbuszról érkezik). Mivel az aritmetikai egység kombinációs áramkör, a kimenetét nem lehet közvetlenül a buszra kapcsolni, mert ez visszacsatolást hozna létre. Ennek megakadályozása végett van szükség a Z regiszterre, amely az eredményt ideiglenesen tárolja. Ezek a regiszterek nem állnak a programozó rendelkezésére.

Példaként, az alábbi táblázat két alciklus elemi műveleteinek a leírását, valamint a megfelelő vezérlési jeleket adja meg:

Alciklus	Időzítés	Aktív vezérlőjelek
Lehívás	$T_1: \text{MAR} \leftarrow (\text{IP})$ $T_2: \text{Memória} \leftarrow (\text{MAR})$ $T_2: \text{MDR} \leftarrow \text{Memória}$ $\text{IP} \leftarrow (\text{IP}) + 1$ $T_3: \text{IR} \leftarrow (\text{MDR})$	C_3, C_4 C_{13}, C_R C_{12}, C_{15} C_1, C_6
Összeadás	$T_1: Y \leftarrow (\text{MDR})$ $T_2: Z \leftarrow (A) + (Y)$ $T_3: A \leftarrow (Z)$	C_6, C_9 $C_8, C_{10}, C_{\text{ADD}}$ C_7, C_{11}

Az utasítás-lehívási ciklus mindig hasonló módon zajlik, de egy-egy utasításnak a végrehajtási ciklusa más-más elemi műveletsorozatot igényel, amit a végrehajtó egység az utasítás műveleti jelrészének a vizsgálata alapján határoz meg.

Az elemi műveletek sorozatának időegységekhez való hozzárendelésénél *két szabályt* kell figyelembe venni:

- az elemi lépéseknek az események helyes sorrendjét kell követniük (például a címküldés meg kell előzze a memóriából való adat beolvasását);
- a konfliktusokat el kell kerülni (például nem lehet egy időben ugyanabba a regiszterbe írni és onnan olvasni).

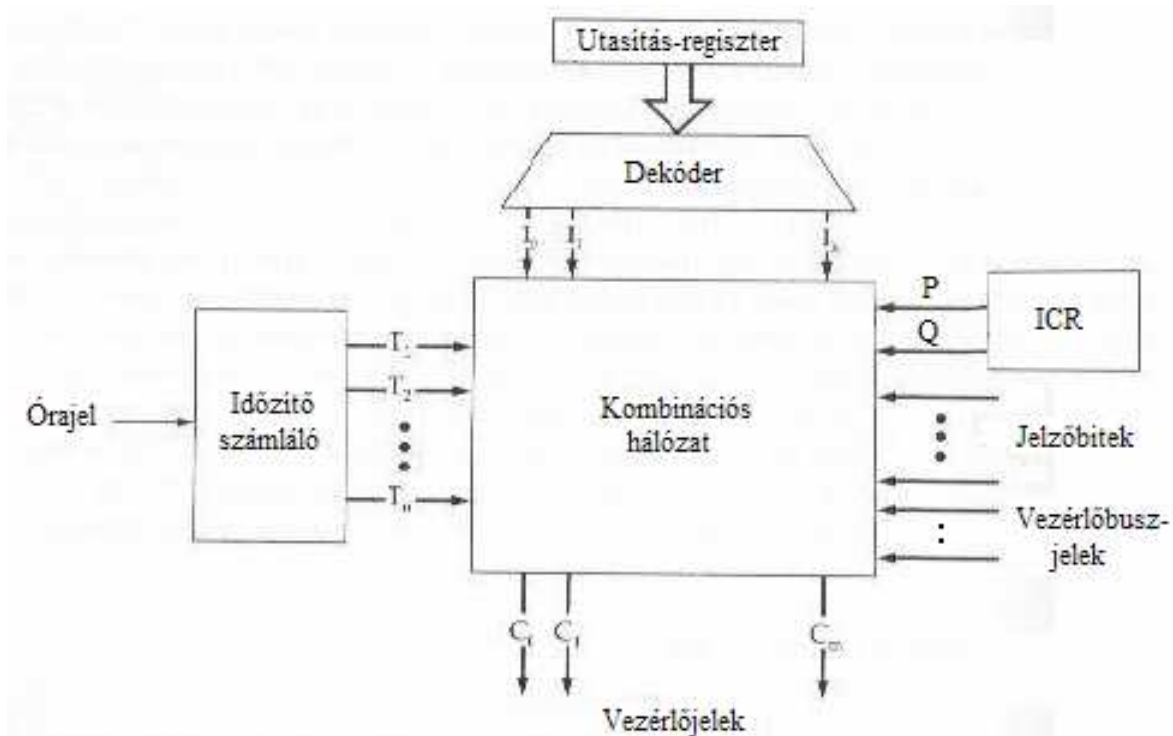
A vezérlőegység implementálására alapvetően két technikát alkalmaznak:

- **Huzalozott** vezérlőegység;
- **Mikroprogramozott** vezérlőegység.

A huzalozott vezérlőegység fő előnye a nagyobb sebességben van, de a mikroprogramozott vezérlőegységnek egyszerűbb az áramköri megvalósítása. Ezért a CISC processzoroknál a mikroprogramozott megoldást alkalmazzák, szemben a RISC processzorok egyszerűbb utasításai számára alkalmasabb huzalozott technikával.

4.2. Huzalozott vezérlőegység

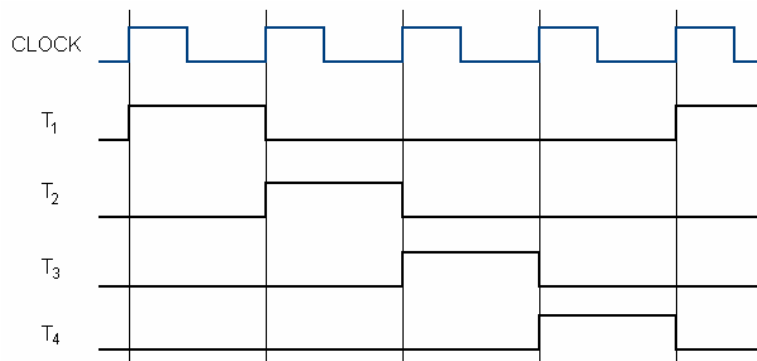
A huzalozott vezérlőegység központi eleme egy kombinációs hálózat, amely a végrehajtandó utasítás által meghatározott bemeneteket alakítja át vezérlőjelekké a kimenetein. Egy ilyen vezérlőegység felépítését mutatja az alábbi ábra:



A vezérlőegység a végrehajtási ciklusban az **utasítás-regiszterben** lévő utasítás **műveleti kódja alapján** kell kibocsásson különböző vezérléseket. Az implementálás egyszerűsítése érdekében minden műveleti kód esetében a kombinációs hálózat egy-egy külön bemenetét kell aktiválni. Ezt a feladatot látja el a **dekóder** áramkör, amely minden bementi kód számára egy kimeneti jelet állít elő. Egy n bemenetű dekódernek 2^n kimenete van, amelyből csak az az egy aktív, amelyet a bemeneti bitkombinációval választottak ki. Példaként egy 3 bemenetű és 8 kimenetű dekóder igazságtáblázatát mutatja a következő ábra:

I2	I1	I0	O0	O1	O2	O3	O4	O5	O6	O7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

A periódikus órajel alapján egy **számláló** állítja elő az elemi műveletek sorrendjét meghatározó T_1 , T_2 stb. impulzusokat:

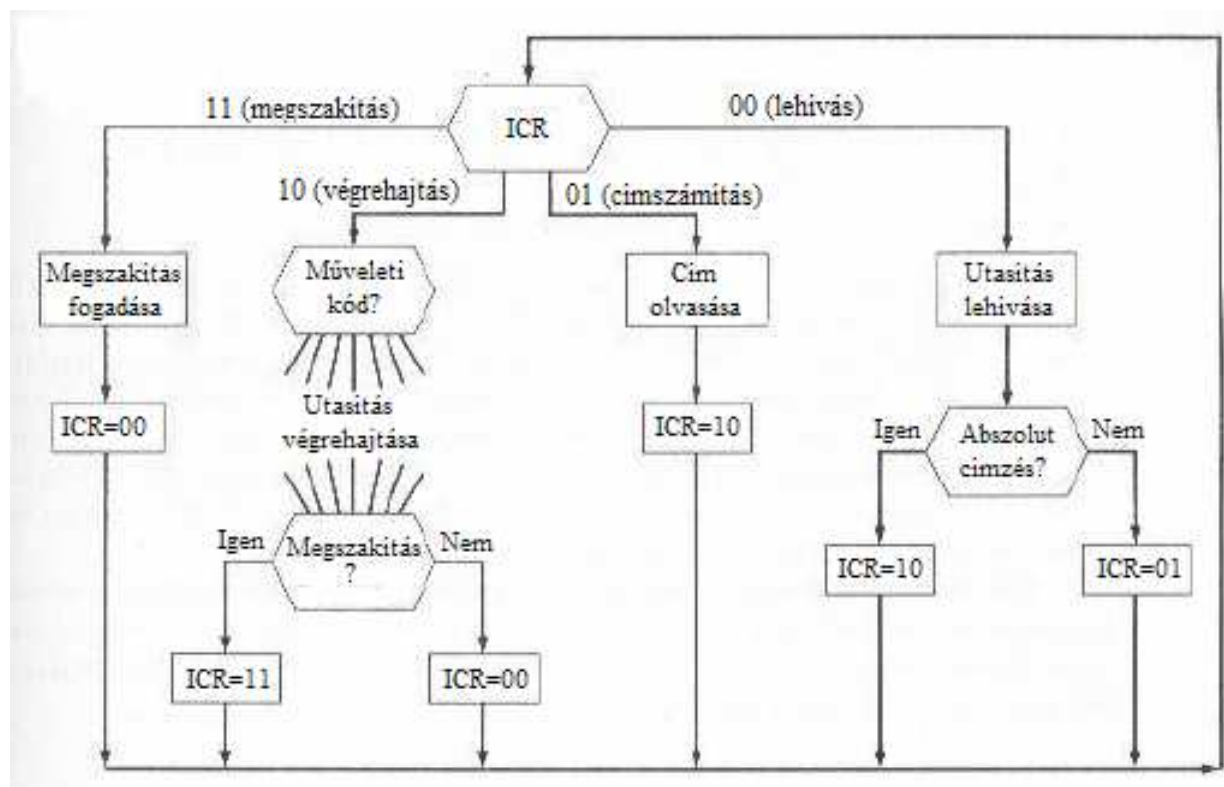


Minden utasításciklus végén a számlálót vissza kell állítani a kezdeti állapotba, hogy a T_1 -et generálja.

Az utasítás lefolyása szempontjából helyes vezérlőjelek előállításához szükséges ismerni, hogy az utasításciklus melyik alciklusa következik. Ezt a célt szolgálja az ábrán az **utasításciklus fázisregisztere** (ICP- *Instruction Cycle Phase*). Az egy egyszerű esetben lehet egy 2-bites (P és Q) regiszter, melynek állapotai például az alábbi alciklusoknak felelnek meg:

PQ = 00	Utasításlehívási ciklus
PQ = 01	Címszámítási ciklus
PQ = 10	Végrehajtási ciklus
PQ = 11	Megszakítási ciklus

A példaként használt ICP működését egy folyamatábrával lehet leírni:



Minden alciklus után a vezérlőegységnek elő kell állítani az ICP megfelelő állapotát, amely meghatározza a következő végrehajtandó alciklust.

Miután minden alciklus számára elkészül a szükséges elemi műveletek időrendi specifikálása, fel lehet írni a Boole-algebra segítségével a **vezérlőjelek egyenleteit**. Például, ha a C_3 vezérlőjelet aktiválni kell minden lehívási és a címszámítási ciklus második időegységében, valamint a MOV, ADD és AND utasítások végrehajtási ciklusának harmadik időegységében, az egyenlete a következő lesz:

$$C_3 = \overline{P} \cdot \overline{Q} \cdot T_2 + \overline{P} \cdot Q \cdot T_2 + P \cdot \overline{Q} \cdot (\text{MOV} + \text{ADD} + \text{AND}) \cdot T_3$$

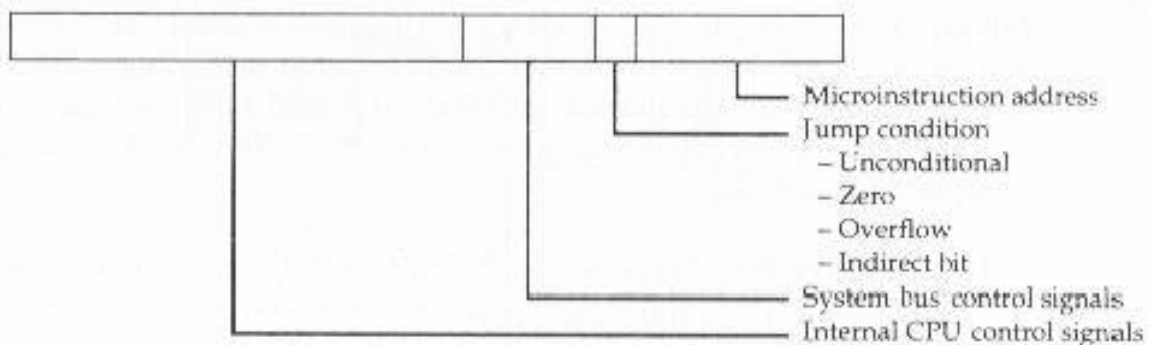
Miután az összes jel számára felírtuk a függvényeket, el lehet végezni ezek alapján a **kombinációs hálózat** logikai kapukkal való implementálását.

4.3. Mikroprogramozott vezérlőegység

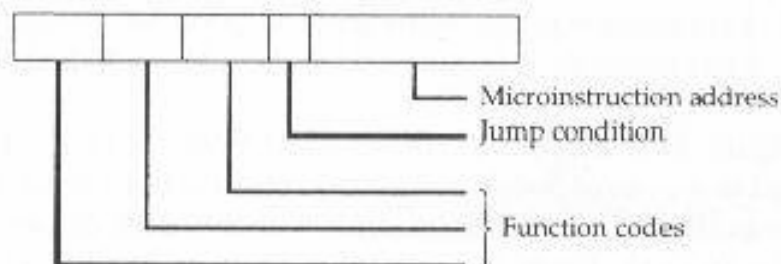
A mikroprogramozott vezérlőegység logikai működését egy **mikroprogram** valósítja meg, amely mikroutasítások sorozatából áll. Minden **mikroutasítás** egy elemi műveletet specifikál, így a végrehajtása a művelethez tartozó vezérlőjelek aktiválását eredményezi.

A mikroutasítások kétféleképpen lehet megvalósítani (lásd az ábrát):

- **Horizontális mikroutasítás** – a mikroutasítás mindegyik bitje egy-egy vezérlőjel bináris alakját jelenti (ha „1”, akkor a vezérlőjel aktív, ha „0”, akkor nem);
- **Vertikális mikroutasítás** – a mikroutasítás a végrehajtandó elemi művelet kódját tartalmazza.

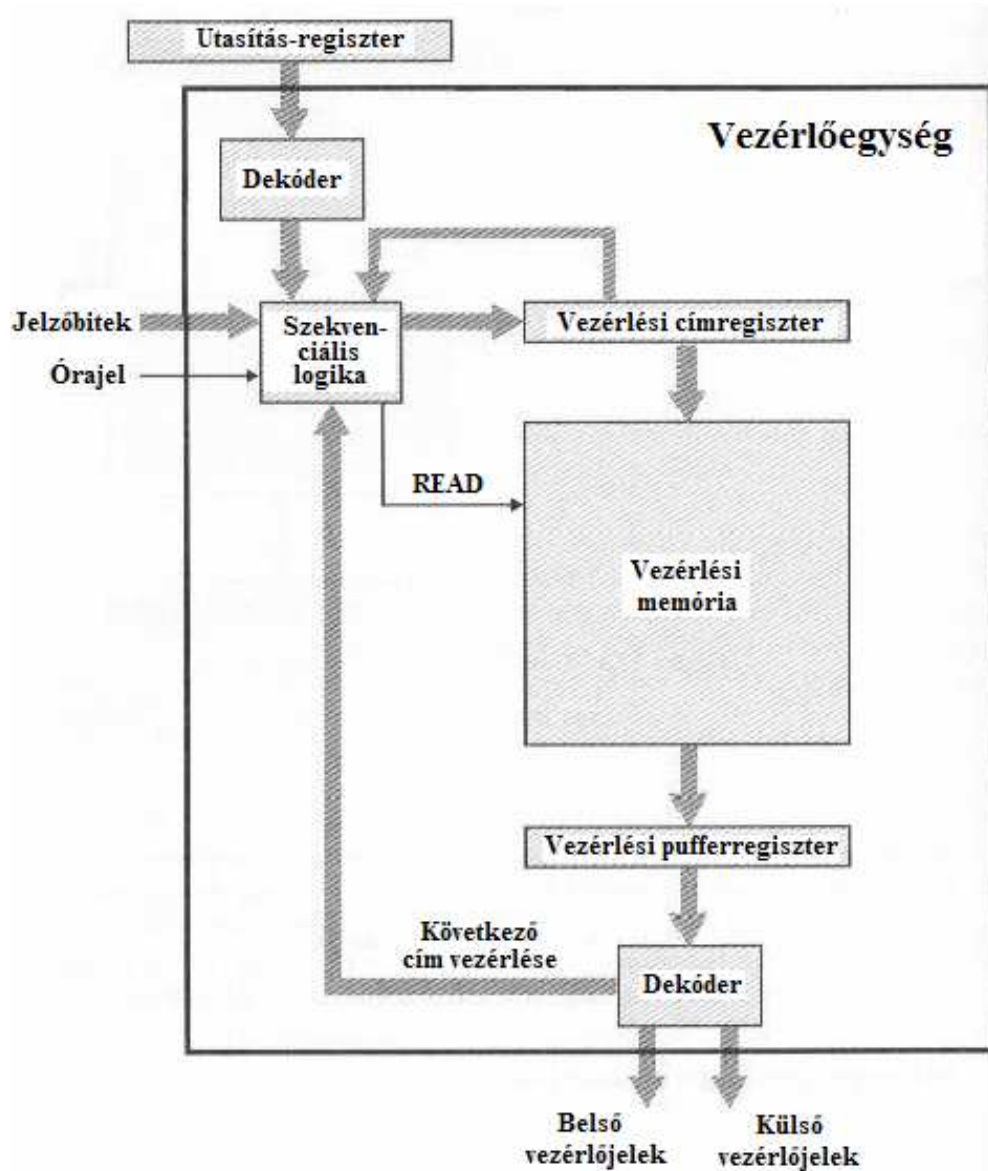


a) Horizontális mikroutasítás

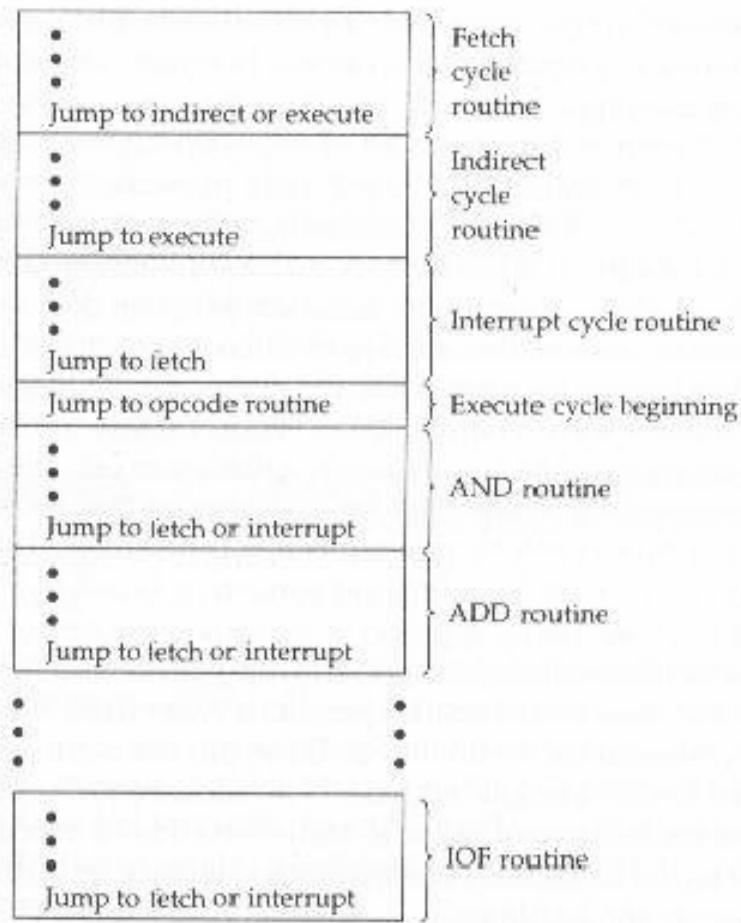


b) Vertikális mikroutasítás

A mikroprogramozott vezérlőegység elvi felépítése a következő:



A *vezérlési memória* egy ROM áramkör, amely a mikroutasítások sorozatát tárolja, szubrutinokba rendezve, amint az alábbi ábra mutatja. Mindegyik alciklusnak egy szubrutin felel meg, a végén egy ugró utasítással a következő végrehajtandó rutinra. A végrehajtási ciklus elején van egy speciális rutin, amely az aktuális műveleti kód alapján választja ki a gépi szintű utasításnak megfelelő rutint (ADD, AND, MOV stb.).



A következő kiolvasandó mikroutasítás címét a *vezérlési címregiszter* tartalmazza. A kiolvasott mikroutasítás a *vezérlési pufferregiszterbe* kerül. A kimenetén lévő dekóderre csak a vertikális mikroutasítások esetében van szükség, mert a horizontális mikroutasítások már a vezérlőjeleknek megfelelő biteket tartalmazzák. A másik, az utasításregiszter kimenetén lévő dekóder, a gépi szintű utasítás műveleti kódját alakítja át vezérlési memóriabeli címmé.

A vezérlőegység működése *minden órajelütemben* a következő:

1. A szekvenciális logika kibocsátja az olvasási parancsot (READ).
2. A megcímzett mikroutasítás bekerül a pufferregiszterbe.
3. A pufferregiszter tartalma előállítja a vezérlőjeleket és a következő mikroutasításra vonatkozó adatot.
4. A következő mikroutasításra vonatkozó adat és az FLAGS regiszter állapotjelzői alapján a szekvenciális logika meghatározza a következő mikroutasítás címét, a következő lehetőségekből:
 - a vezérlési címregiszter tartalmát +1-gyel megnöveli,
 - az ugró (JUMP) mikroutasításban megadott címet tölti be, ezzel egy másik rutinra ugorva;
 - az utasításregiszterben lévő műveleti kódnak megfelelő rutin címét tölti be, elindítva ennek az utasításnak a végrehajtási alciklusát.

A mikroprogramozott vezérlőegységnek két alapfeladatot kell ellátni:

- A mikroutasítások *szekvenciálását*, azaz meghatározni a következő mikroutasítás címét;
- A mikroutasítások *végrehajtását*, vagyis generálni a szükséges vezérlőjeleket.

Mindkét feladat megoldása szoros összefüggésben van a mikroutasítások szerkezetével.

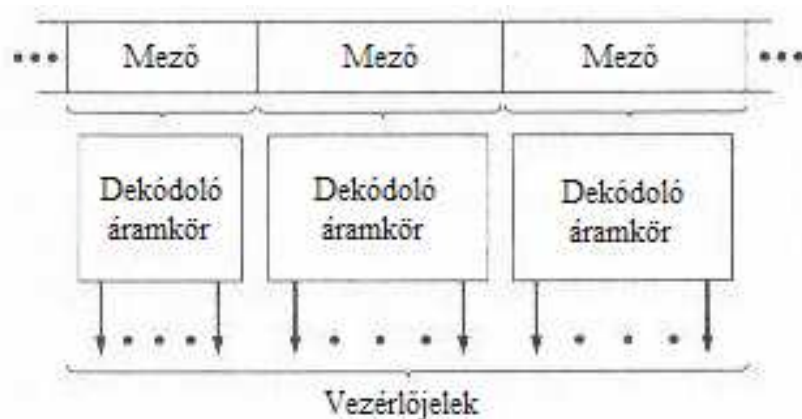
4.4. A mikroutasítások kódolása

A tiszta horizontális, illetve vertikális utasításszerkezet két végletet jelent: ez egyiknél minden vezérlőjelet egy külön bit jellemez, a másikra egy magas fokon becsomagolt formátum a jellemző. A gyakorlatban minden vezérlőegység tervezésénél valamilyen fokig szokás a mikroutasításokat kódolni, hogy csökkenjen a vezérlőmemória szélessége, és egyszerűsödjön a mikroprogramozás feladata.

A kódolt mikroutasítások tervezésének menete a következő:

1. Szervezzük meg a szerkezetet mezőkre bontva úgy, hogy a mezőkre jellemző műveletek vezérlőjelei egyszerre is aktiválódhassanak. Így N mezővel N egyidejű eseményt lehet meghatározni.
2. Határozzuk meg minden mezőben a specifikálható, egymást kizáró műveleteket úgy, hogy azokból egyszerre csak egy lehessen aktív. Ha egy mező M bitet tartalmaz, akkor ezzel maximum 2^M egymást kizáró eseményt lehet kódolni.

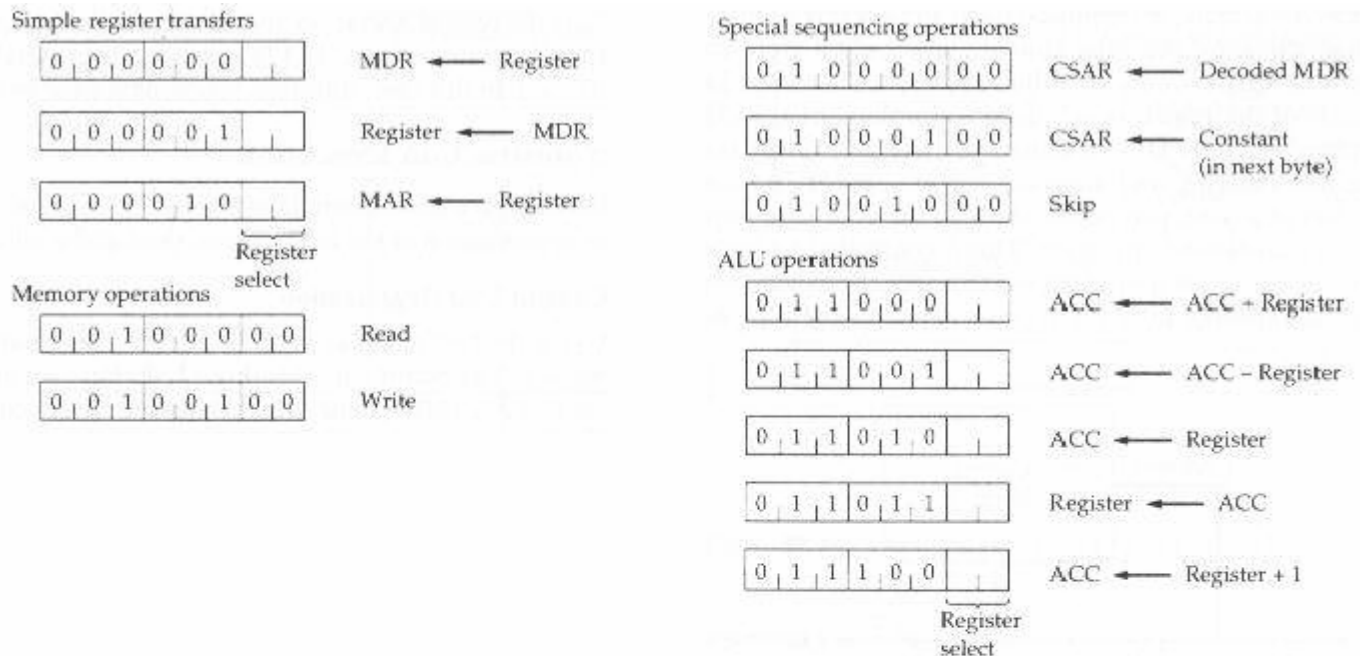
A létrejött mikroutasítások mindegyik mezője tartalmaz egy kódot, amelyet dekódolva létrejönnek a vezérlőjelek:



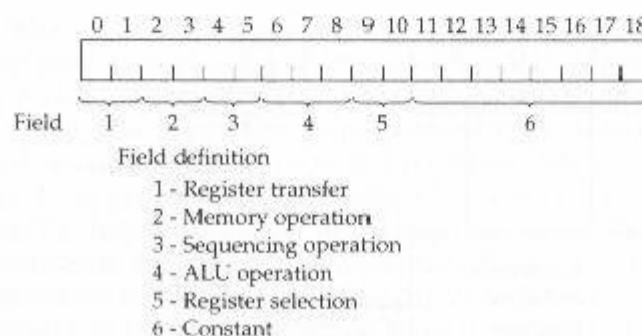
A mikroutasítások kódolásának két technikája van:

- **Funkcionális kódolás** – a gép minden egyes funkciójának egy-egy mezőt tartunk fel;
- **Erőforrás szerinti kódolás** – a gép minden erőforrásához rendelünk egy-egy mezőt.

Erre a két technikára mutat példákat az alábbi ábra:



a) Funkcionális kódolás



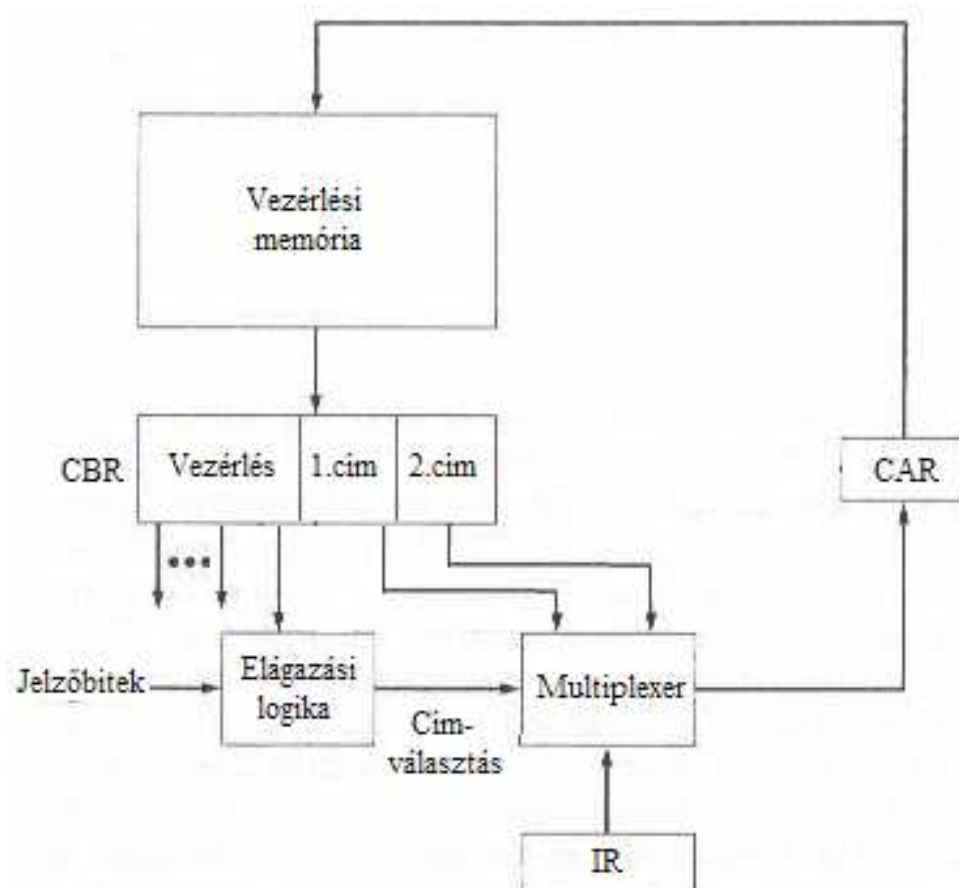
b) Erőforrás szerinti kódolás

4.5. A mikroutasítások szekvenciálása

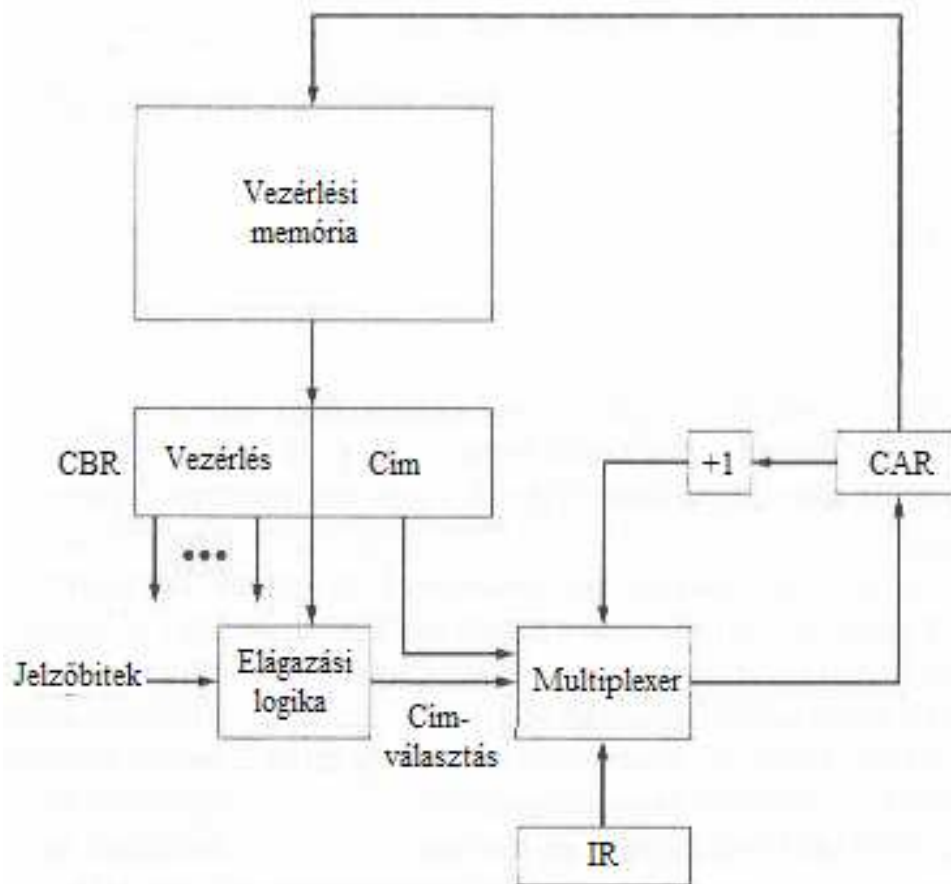
A következő végrehajtandó mikroutasítás számára egy vezérlőmemóriabeli címet kell előállítani az aktuális mikroutasítás, a jelzőbitek és az utasítás-regiszter tartalma alapján. A mikroutasítás szerkezetében lévő címinformációk alapján, az erre használt technikák három csoportba foglalhatók:

- Két címmezős szerkezet;
- Egy címmezős szerkezet;
- Változó szerkezet.

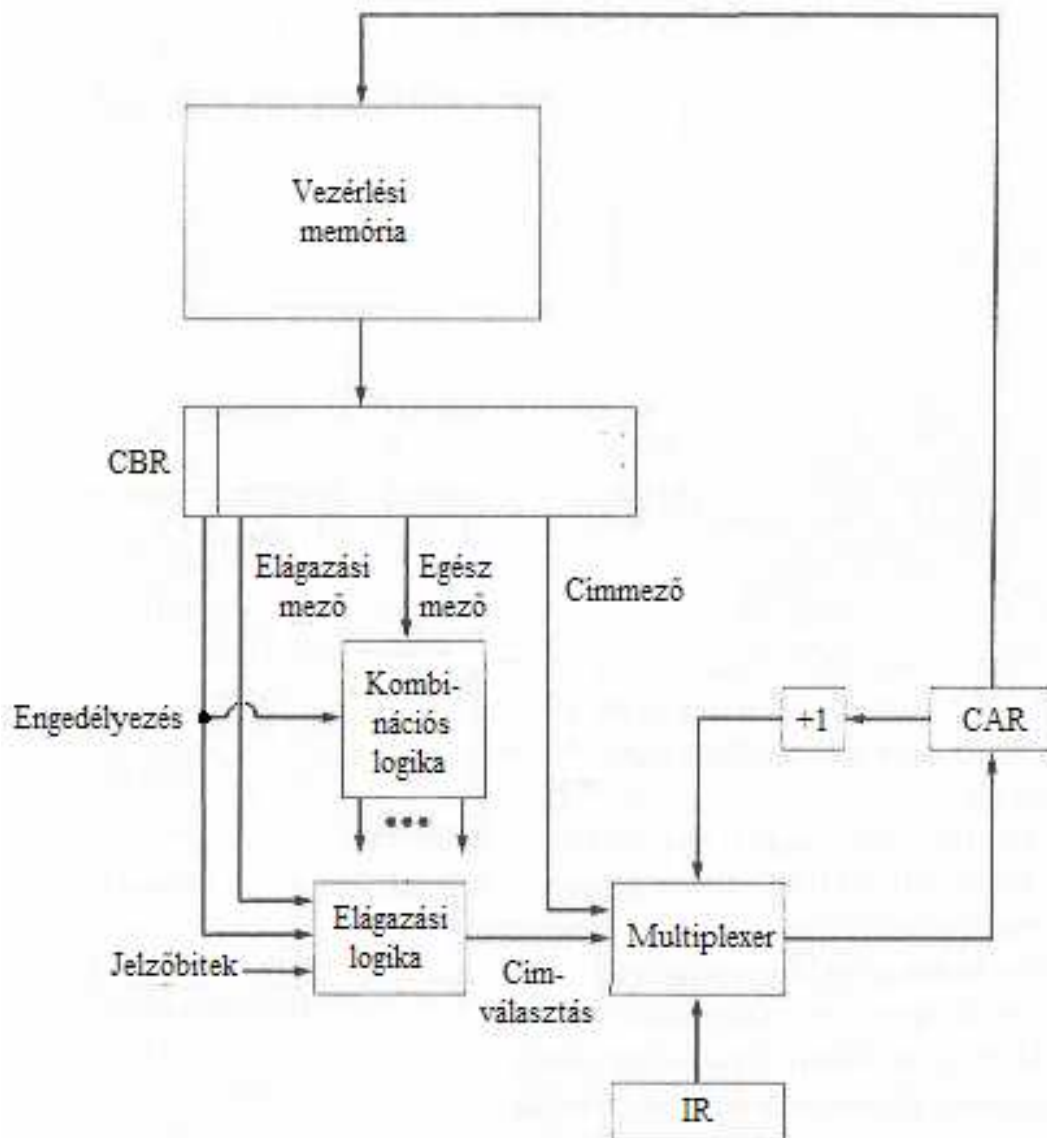
A legegyszerűbb megoldás a **kétcímes szerkezet** (lásd az alábbi ábrát), de hátránya a hosszú mikroutasítás. Az egyik cím a következő mikroutasítás címe, a másik a feltételes elágazásnál érvényes cím. Egy multiplexáló áramkör továbbítja a vezérlési címregiszterbe a műveleti kódot vagy a két cím egyikét. A kiválasztást az elágazási logika végzi el, a mikroutasítás vezérlőrésze és a jelzőbitek alapján.



A takarékosabb, **egycímes szerkezetnél**, a soron következő utasítás címét – ugrás hiányában – az aktuális utasítás címének inkrementálásával kapjuk meg:



Az előző megoldás sem a leghatékonyabb, mert a mikroutasításban fenntartott címmezőre csak az ugrási utasításokban van szükség. **A változó szerkezet** esetében, a mikroutasítás egy külön bitje határozza meg, hogy melyik formátumról van szó: az egyik formátumban a fennmaradó bitek a vezérlőjeleket aktiválják, a másikban egy részük az elágazási logikát vezérli, a többi az ugrási címet tartalmazza (ez utóbbi esetben egy órajelütemre kimarad a vezérlőjelek generálása):



Az eddig bemutatott **explicit** technikákon kívül léteznek **implicit** címgenerálási technikák is:

- A gépi utasítás kódja jelenti a mikroutasítás címét;
- A mikroutasítás címmezőjének részein végrehajtott művelettel (például összeadással) állítjuk elő a következő mikroutasítás címét;
- Egy mikroszubrutin végén a visszatérési címet egy – a vezérlőegységben belüli – regiszterből vagy veremből olvassuk ki.

A mikroprogramozott módon implementált vezérlőegység lehetővé teszi az **emulációt**. Ez alatt azt értjük, hogy egy számítógépen – megfelelő mikroporgram segítségével –, *lehetővé tesszük egy másig gépre írt, annak utasításait használó program végrehajtását.*

KÉRDÉSEK, TÉTELEK

- 4.1. Melyek a vezérlőegység alapfeladatai?
- 4.2. Vázolja fel a huzalozott vezérlőegység felépítését, és magyarázza el a működését.
- 4.3. Mi a különbség a horizontális és a vertikális mikroutasítás között? Mutassa be egy ábrán!
- 4.4. Vázolja fel a mikroprogramozott vezérlőegység felépítését, és magyarázza el a működését.
- 4.5. Mi a kódolt mikroutasítások tervezésének a menete? Milyen fajta kódolást alkalmazhatunk?
- 4.6. Hogy határozzuk meg a következő mikroutasítás címét különböző szerkezetű mikroutasítások (egy címmezős, két címmezős, változó formátumú) esetében?

5. MEMÓRIA-HIERARCHIA TERVEZÉSI KÉRDÉSEI.

5.1. A központi tár címzése

A központi tárhoz való hozzáférés érdekében a processzor az illető **rekesz címét küldi ki a címbuszon**. A címbusz szélessége meghatározza a közvetlenül megcímezhető rekeszek számát, azaz a **memória címtartományát** (m bites cím esetében a címtartomány mérete 2^m). A mikroszámítógépekben minden rekesz 8 bitet (1 bájt) tartalmaz.

A címtartomány egy részét – nem feltétlenül az egészet – foglalják el a rendszerbe fizikailag bekötött memória-áramkörök. Ebben a fizikailag létező tárban egy memóriarekesz megcímezése két feladatot feltételez:

- az illető rekeszt tartalmazó *csip kiválasztását*
- a *rekesz kiválasztását* a csipen belül.

A csipen belüli rekesz kiválasztása a címbusz legkisebb helyiértékű vonalaival történik, amelyeket az áramkör címbemeneteire kell kötni. Ezeknek a számát az illető memória-áramkör kapacitása határozza meg. Például egy 1 KB-os áramkörnek 10 címbemenete van ($2^{10} = 1\text{ K}$).

A címbusz többi vonalát a memóriacsipek kiválasztására használjuk. Ezek határozzák meg az áramkörök által elfoglalt fizikai memória helyét a címtartományon belül.

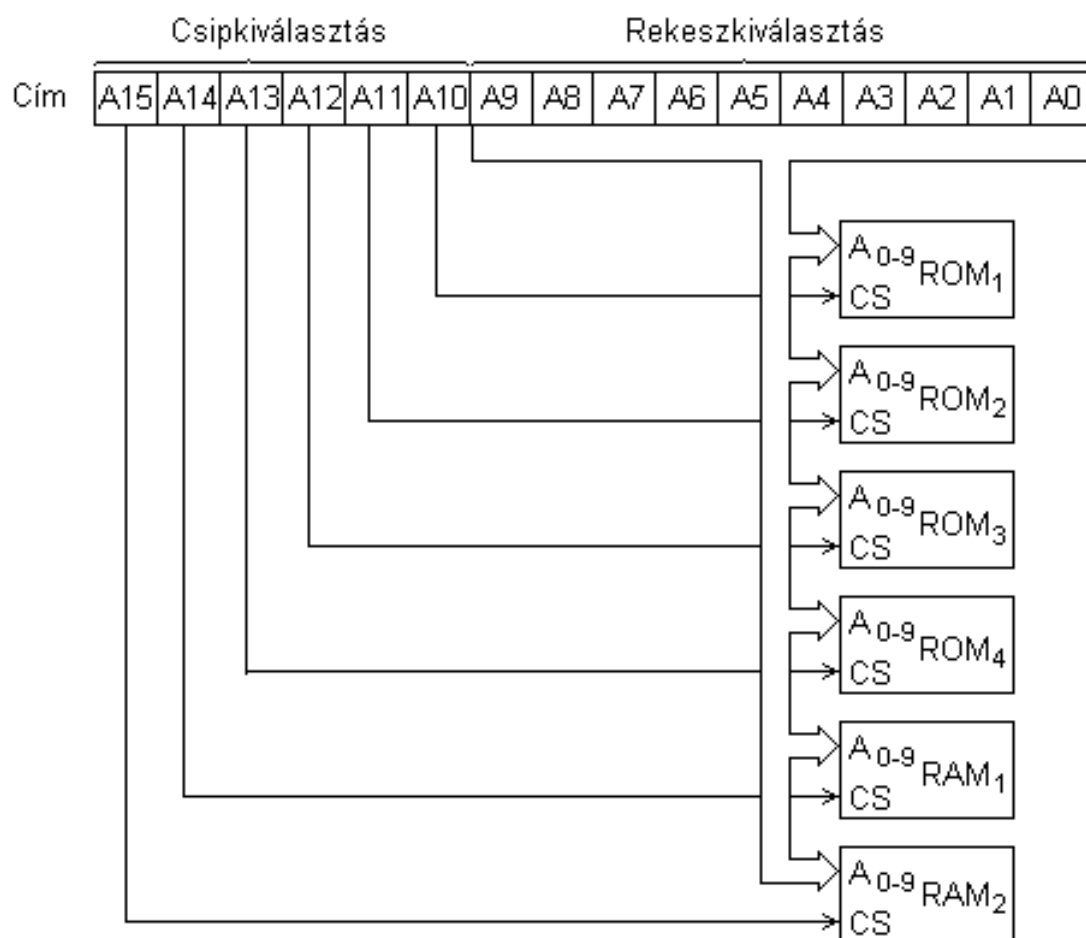
A csipkiválasztásnak két technikája van:

- **lineáris** szelekció
- **dekódolt** szelekció

a) Lineáris szelekció

Ebben az esetben **minden áramkör kiválasztására egy-egy külön címvonalat használunk fel**. Ezt az egyszerű módszert csak kevés memória-áramkört tartalmazó rendszerekben tudjuk használni, ahol a megmaradt magasabb helyiértékű címvonalak száma nem kisebb az áramkörök számánál.

Az alábbi példa egy 16 bites címbusszal rendelkező rendszert mutat be, amelyben 1 KB-os memóriacsipek vannak. Ezeknek az A_0 – A_9 -es címbemenetei szolgálnak az áramkörön belül egy rekesz kiválasztására, míg a CS (*Chip Select*) bemenetnek a feladata az áramkör kiválasztása:



A fizikai memóriaterület elhelyezkedését a processzor címtartományában **memóriatérképnek** (*memory map*) hívjuk. A fenti példa esetében a memóriatérkép az alábbi ábrán látható (a címek 16-os számrendszerben vannak megadva):

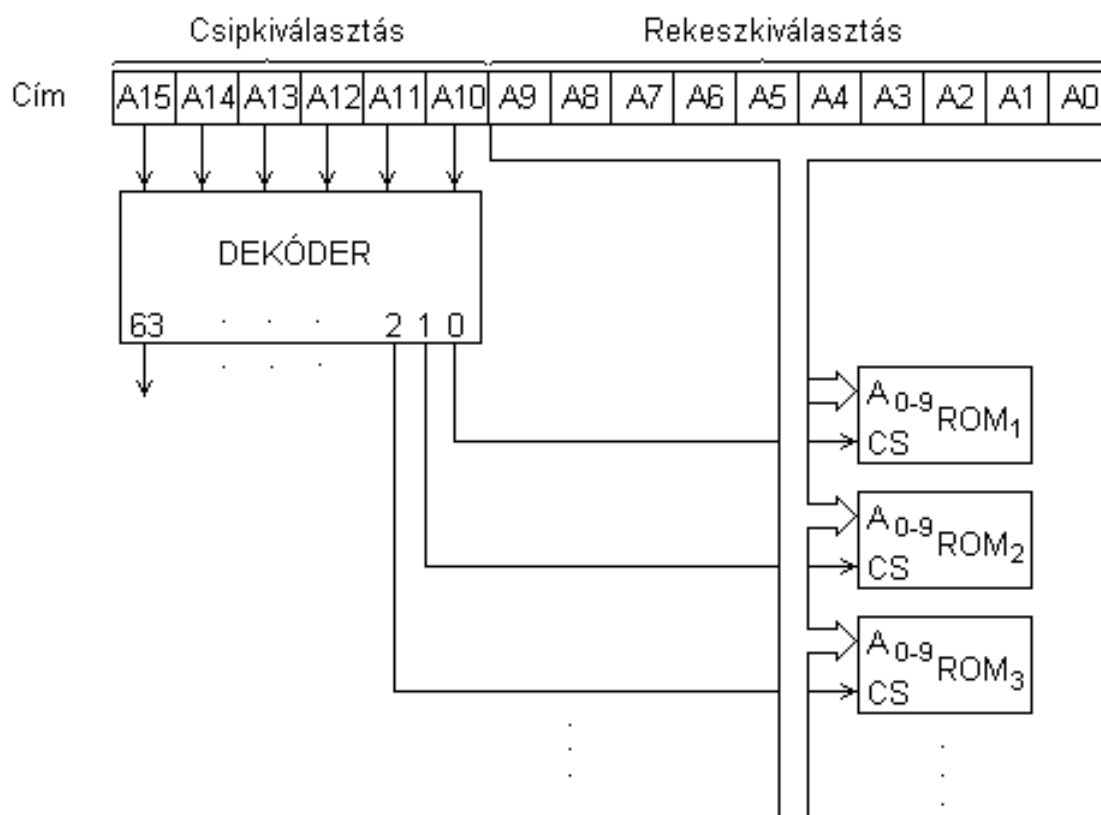
Cím	Főtár
0000 _H – 03FF _H	1 K szabad
0400 _H – 07FF _H	1 K ROM ₁
0800 _H – 0BFF _H	1 K ROM ₂
0C00 _H – 0FFF _H	1 K szabad
1000 _H – 13FF _H	1 K ROM ₃
1400 _H – 1FFF _H	3 K szabad
2000 _H – 23FF _H	1 K ROM ₄
2400 _H – 3FFF _H	7 K szabad
4000 _H – 43FF _H	1 K RAM ₁
4400 _H – 7FFF _H	15 K szabad
8000 _H – 83FF _H	1 K RAM ₂
8400 _H – FFFF _H	31 K szabad

Látható, hogy lineáris szelekcióval nem lehet kitölteni az egész címtérületet, és töredezett memóriaterületet eredményez, ami ennek a módszernek a hátránya.

b) Dekódolt szelekció

Ebben az esetben a csipkiválasztás számára *megmaradt címvonalak dekódolásával* állítjuk elő a memória-áramkörök szelekciós jeleit. Ez a módszer lehetővé teszi az egész címtartomány kihasználását, valamint ezen belül folyamatos memóriaterületeket lehet létrehozni.

Az alábbi ábra az előző példát mutatja be dekódolt szelekció alkalmazásával. Az A10–A15 címbitek dekódolásával létrejön $2^6 = 64$ jel a csipek kiválasztására. Ezek a jelek a 64 KB-os címtartományt 1 KB-os zónákra bontják, amint az a memóriatérképen látható.



Cím	Főtár
0000 _H – 03FF _H	1 K ROM ₁
0400 _H – 07FF _H	1 K ROM ₂
0800 _H – 0BFF _H	1 K ROM ₃
.	.
.	.
.	.

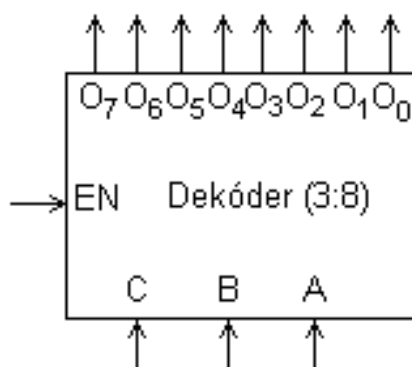
A dekódoló logikát meg lehet építeni különálló kapukból, vagy dekóder áramkör alkalmazásával. A *csipkiválasztó jelek egyenleteit* – az előbbi példánál maradva – a következőképpen írhatjuk fel:

$$CS_1 = \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot \overline{A_{10}}$$

$$CS_2 = \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot \overline{A_{11}} \cdot A_{10}$$

$$CS_3 = \overline{A_{15}} \cdot \overline{A_{14}} \cdot \overline{A_{13}} \cdot \overline{A_{12}} \cdot A_{11} \cdot \overline{A_{10}}$$

Ezen egyenletek alapján készíthetjük el a dekódoló logikát, vagy használhatunk kész **dekódoló áramkört**. Egy ilyen 3 bemenetű és $2^3 = 8$ kimenetű áramkörnek a logikai vázlatát és a működési táblázatát tartalmazza az alábbi ábra. Az EN (*Enable*) bemenet a dekódolási funkció engedélyezését jelenti.

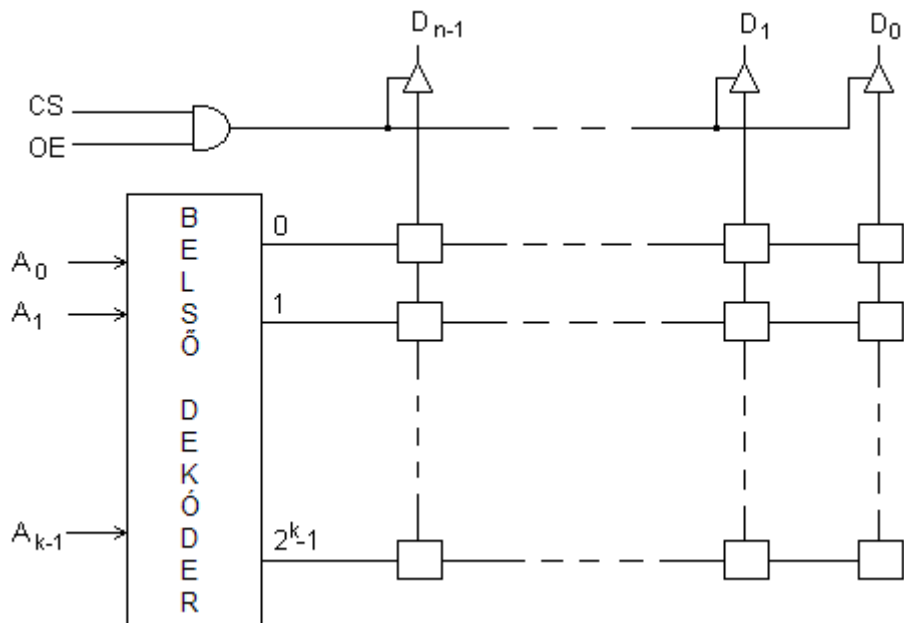


EN	C	B	A	O ₇	O ₆	O ₅	O ₄	O ₃	O ₂	O ₁	O ₀
0	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

2. A memória-áramkörök bekötésének elve

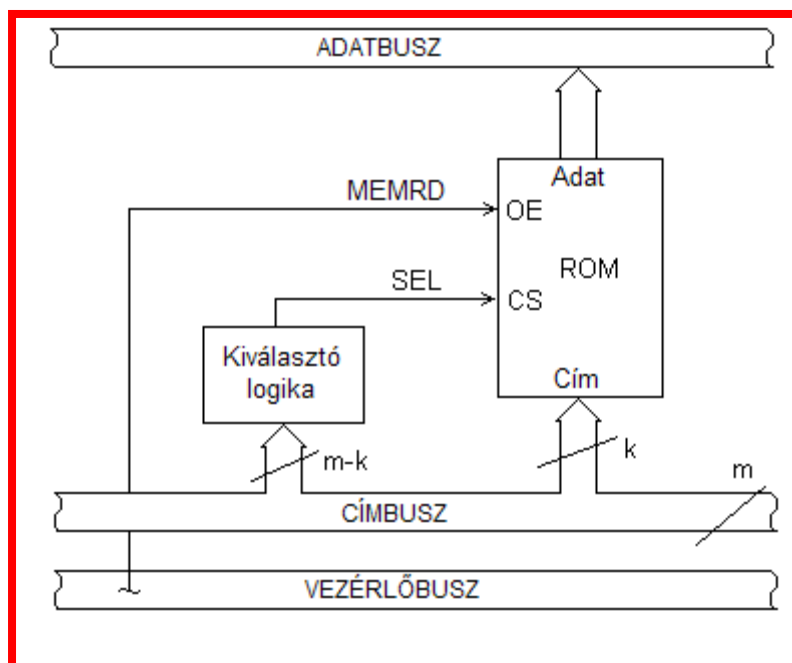
a) ROM memóriák

Egy ROM áramkör memóriacellái mátrix alakban vannak elrendezve a csip belsejében (lásd az ábrát). Minden egyes cella egy bitet tárol. A k a címbemenettel ($A_0 - A_{k-1}$) lehet aktiválni a mátrix egyik sorát, és így kiválasztani az egyiket az összesen 2^k n -bites rekesz közül. Az áramkör kapacitása tehát $2^k \times n$ bit.



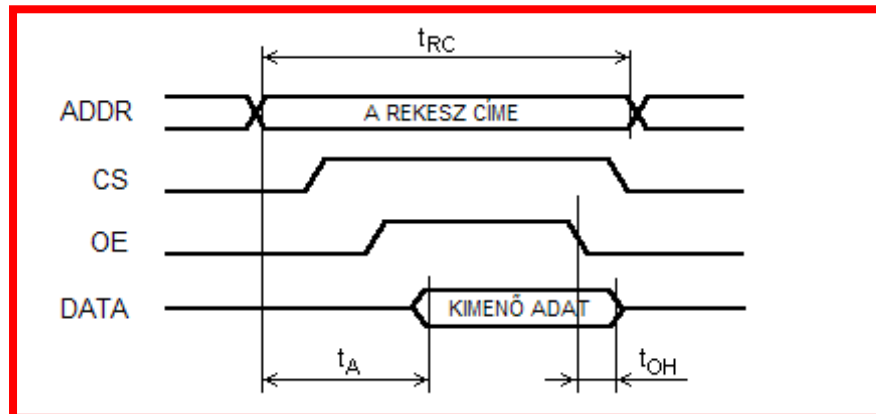
A mátrix oszlopai 3-állapotú puffer meghajtókon keresztül csatlakoznak az adatkimenetekre ($D_0 - D_{n-1}$). Ezáltal lehetővé válik a kimenetek közvetlen csatlakoztatása az adatbuszra. Az adat továbbítása a kimeneteket engedélyező OE (*Output Enable*) jel hatására történik, feltéve, hogy az áramkör szelektálva van ($CS = 1$).

Egy ROM áramkör elvi bekötését a rendszer buszaira a következő ábra szemlélteti:



Miután a processzor a címbuszra helyezte a kiolvasandó rekesz m bites címét, ez a cím kettéválk: az alsó k bit közvetlenül a 2^k kapacitású csip címbemeneteire kerül, míg a megmaradó bitekből a kiválasztó logika generálja a csip szelekciós jelét (SEL), a már ismert technikák egyikével (lineáris vagy dekódolt szelekció). A kiolvasott adatok a memóriaolvasási vezérlőjel (MEMRD) hatására kerülnek az adatbuszra.

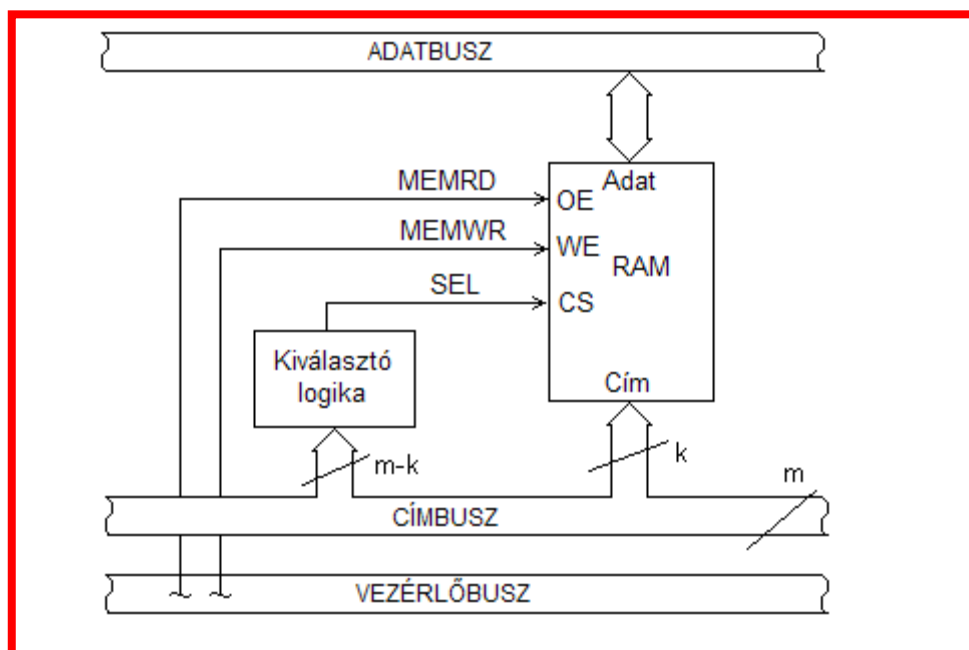
Minden memória-áramkör fontos jellemzője a **hozzáférési idő** (t_A – access time), amely a címnek a csip bemeneteire való helyezésének pillanatától az adatok kimeneteken való megjelenéséig tart. A hozzáférési időt az áramkör olvasási ciklusát leíró idődiagram mutatja:



Az olvasási ciklus ideje alatt (t_{RC} – Read Cycle) a címnek stabilan meg kell maradni a memória bemenetein. A kimenetek letiltása után az adat még egy tartási ideig (t_{OH} – Output Hold) elérhető, és így átvehető a processzor által az adatbuszról. A processzor működését egy olvasási ciklusban össze kell hangolni a használt memóriára jellemző olvasási ciklussal.

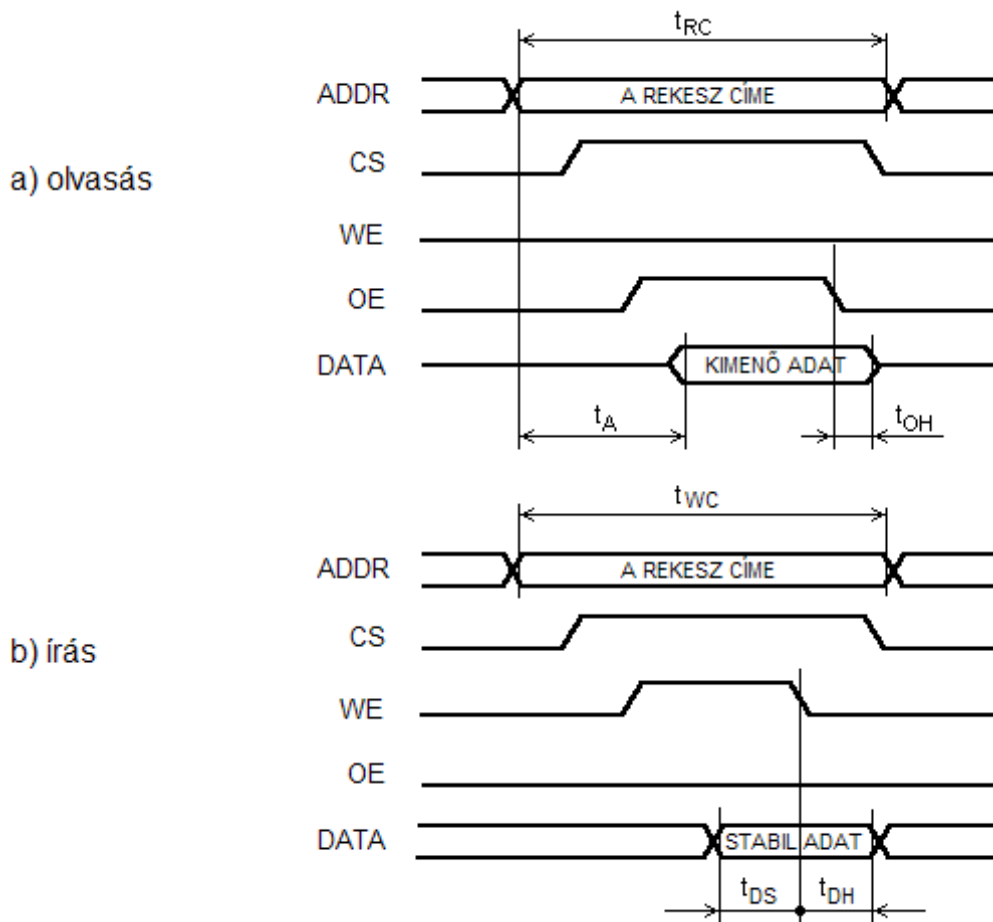
b) RAM memóriák

A statikus RAM áramkör buszrendszerhez kapcsolását mutatja a következő ábra:



A bekötés módja hasonló a ROM áramköréhez, de itt van még egy vezérlőbemenet, amely az írást teszi lehetővé (WE – *Write Enable*). Erre a bemenetre a processzortól jövő memóriaírási vezérlőjel (MEMWR) kerül, míg a kimenetek megnyitása az olvasási ciklusban történik a MEMRD jellel.

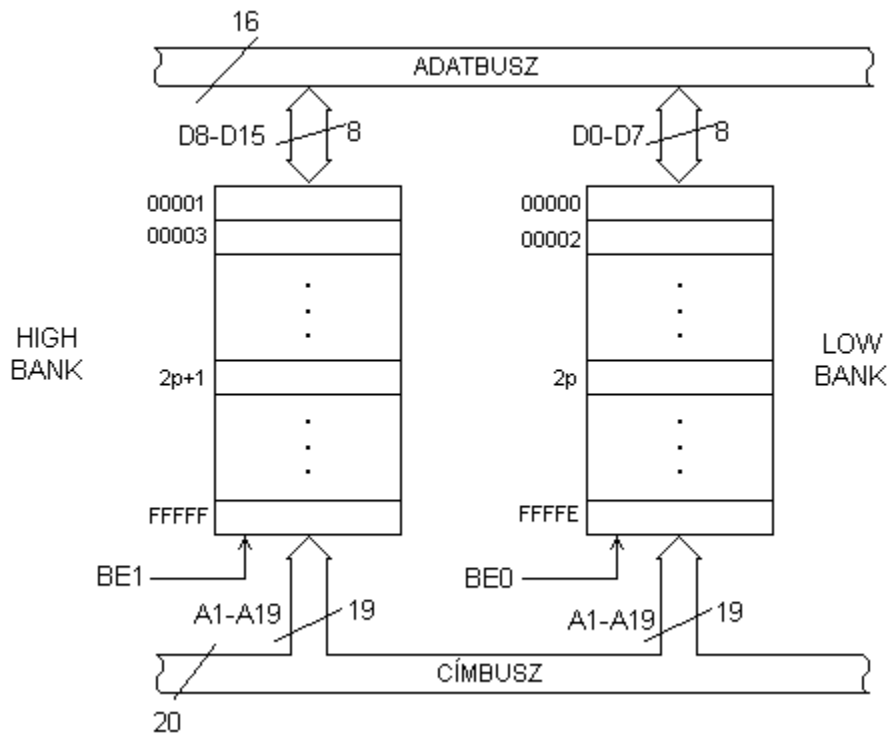
A helyes működést a memória-áramkör idődiagramjai mutatják:



Az írási ciklusban az adat a processzortól érkezik az adatbuszon. Ez stabil kell maradjon a WE jel hátsó éle körül, mert ekkor történik az adat beírása a memóriába. A csip katalógusában megadják az **előkészítési idő** (t_{DS} – *Data Setup*) és a **tartási idő** (t_{DH} – *Data Hold*) minimálisan szükséges értékeit.

A memória-áramkörök bekötésének sajátosságai széles adatbusszal rendelkező rendszerek esetében

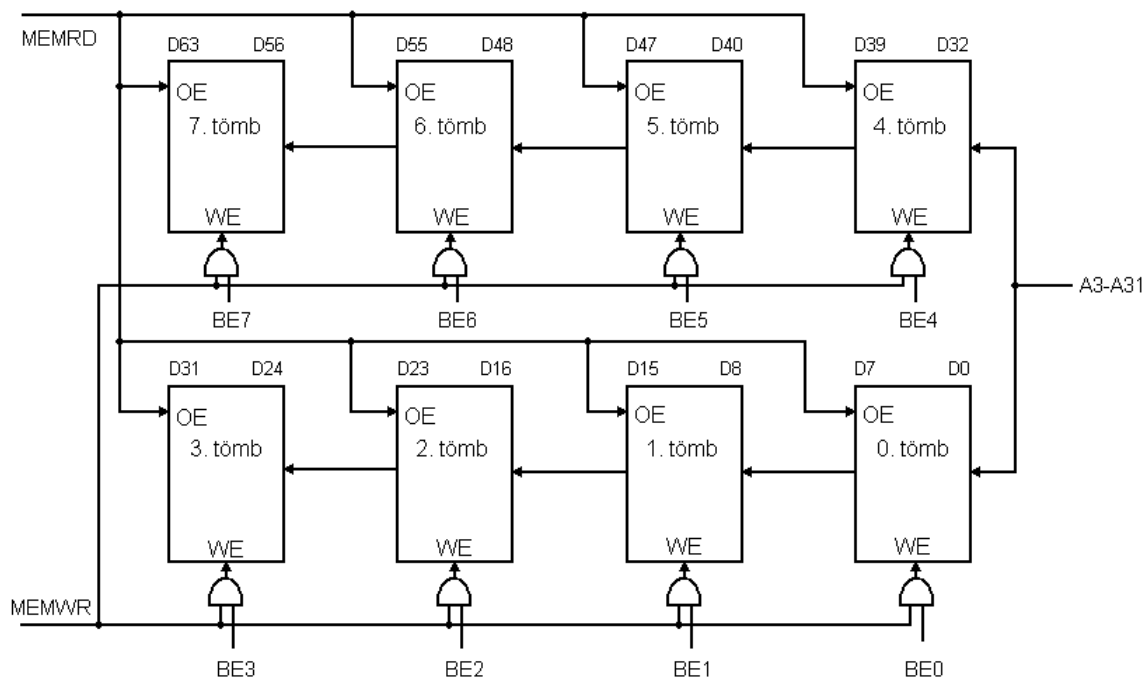
Egy 16 bites adatbusszal rendelkező processzor esetében lehetőség van egyszerre két bájt átvitelére a memóriából. Ennel feltétele, hogy a memóriát két tömbben szervezzük meg, amelyek mindegyike az adatbusz egy-egy 8 bites részéhez kapcsolódik:



Az **alsó tömb** (*low bank*) tartalmazza a páros című rekeszeket ($A0 = 0$), és az adatbusz D0–D7 vonalaihoz csatlakozik. A páratlan című rekeszek ($A0 = 1$) a **felső tömbben** (*high bank*) vannak, amelyek az adatbusz D8–D15 vonalaihoz csatlakozik. A tömbök kiválasztására a processzor két **bájt kiválasztó jelet** (BE – *Byte Enable*) küld, ezért az A0 címvonalra nincs is szükség. Három fajta adatátvitel lehetséges:

- 1 bájt átvitele páros címről a D0–D7-en – ekkor $BE1 = 0$, $BE0 = 1$
- 1 bájt átvitele páratlan címről a D8–D15-ön – ekkor $BE1 = 1$, $BE0 = 0$
- 2 bájt átvitele páros címtől kezdődően a D0–D15-ön – ekkor $BE1 = 1$, $BE0 = 1$

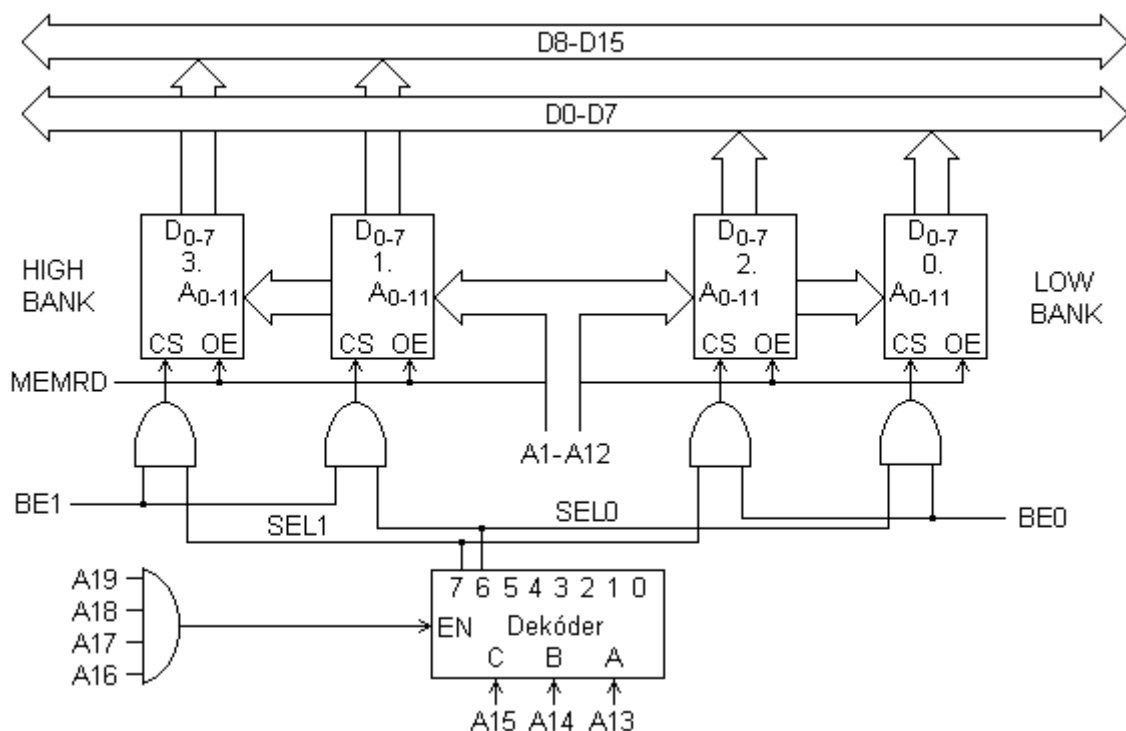
Az előbbi elvet lehet általánosítani 32, illetve 64 bites adatbuszok esetére is. A 64 bites adatbusszal rendelkező rendszerben 8 tömböt kell kialakítani, amint az alábbi ábra mutatja. Itt 8 bájt kiválasztó jel van, ezért az A0–A3 címbitekre nincs szükség (ezeket a processzor belsőleg használja a bájt kiválasztó jelek generálására).



Az előző ábrán figyeljük meg, hogy a bájt kiválasztó jelek az írás engedélyezésében vesznek részt, mivel az olvasás nem destruktív (egyszerre ki lehet olvasni a címterületen szomszédos címen lévő, de mégis külön tömbben helyet foglaló adatokat, még ha a processzornak nincs is mind a 8 bájtra szüksége – írásnál viszont ezt már nem szabadna megtenni).

3. ROM memóriablokk

Az alábbi példa egy 16 KB-os ROM memóriablokkot mutat be, egy 16 bites adatbusszal és 20 bites címbusszal rendelkező rendszer számára. A felhasznált csipek szervezése $4K \times 8$ bit.



A memóriablokk két tömbbe van szervezve, az egyik a busz D0–D7, a másik a D8–D15 vonalaira van kötve. A tömbök kiválasztását a BE0 és BE1 jelek végzik. A memóriablokk által elfoglalt címterületet az alábbi táblázat mutatja, amely az áramkör párok címhatárait tartalmazza:

A19	A18	A17	A16	A15	A14	A13	A12	...	A1	A0	Csip	Szelekció	Címterület
1	1	1	1	1	1	0	0	...	0	0	0. és 1.	SEL0=1	FC000 _H ÷FDFFF _H
1	1	1	1	1	1	0	1	...	1	1			
1	1	1	1	1	1	1	0	...	0	0	2. és 3.	SEL1=1	FE000 _H ÷FFFF _H
1	1	1	1	1	1	1	1	...	1	1			

A páros címeket (A0=0) tartalmazó csipek az alsó tömbben, a páratlant (A0=1) tartalmazók a felsőben vannak. A csipeken belüli rekeszek kiválasztására az A1–A12 címbiteket használjuk, a megmaradt A13–A19 biteket dekódolva állítjuk elő a táblázat sorainak megfelelő SEL0 és SEL1 jeleket. A csipkiválasztó jeleket a következő egyenletek adják meg:

$$CS_0 = SEL0 \cdot BE0$$

$$CS_1 = SEL0 \cdot BE1$$

$$CS_2 = SEL1 \cdot BE0$$

$$CS_3 = SEL1 \cdot BE1$$

(Meg kell jegyezni, hogy mivel csak olvasható memóriáról van szó, nem feltétlenül lenne szükséges a bájt kiválasztó jelek használata.)

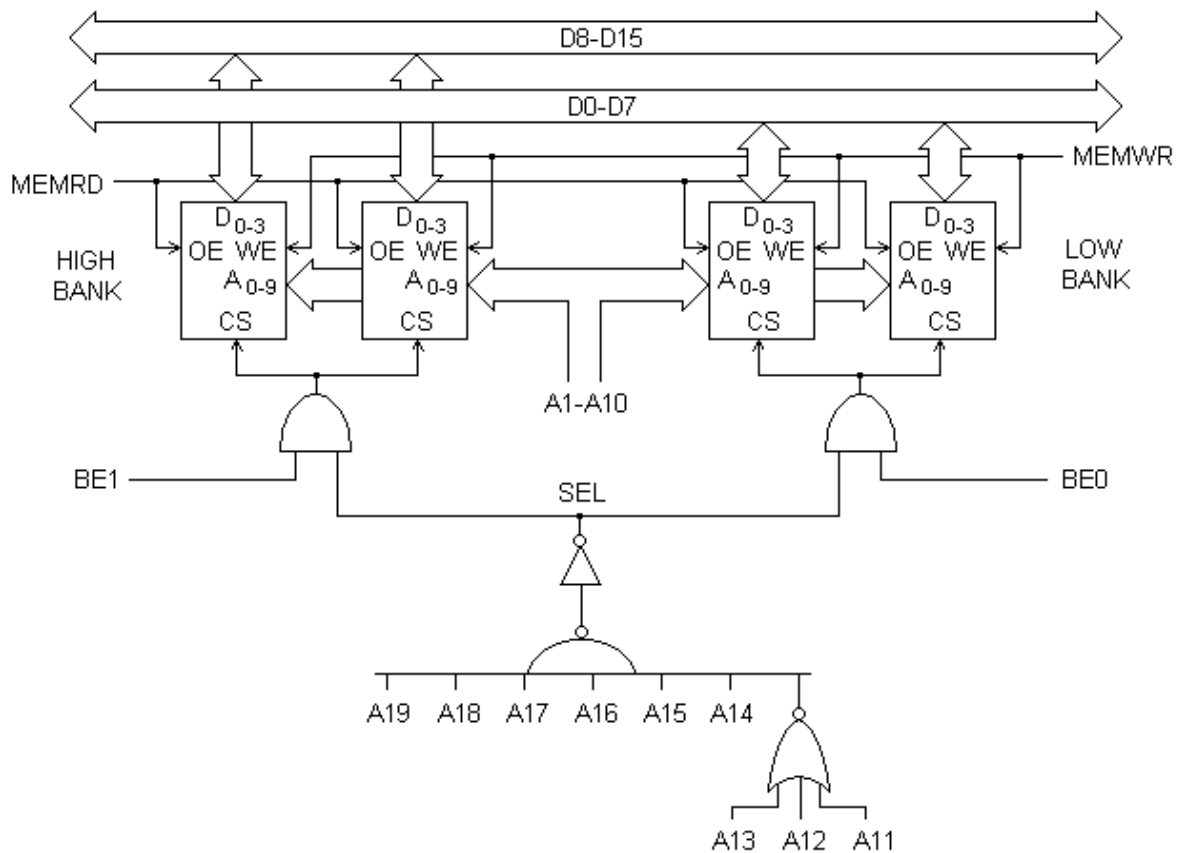
A dekóder bemeneteire kerülnek az A13–A15 vonalak, a további bitek (amelyeknek az értéke változatlan) biztosítják a dekóder engedélyezését::

$$EN = A19 \cdot A18 \cdot A17 \cdot A16$$

A kimeneteket az olvasási ciklusban aktiválódó MEMRD (*Memory Read*) vezérlőjel nyitja meg.

4. RAM memóriablokk

A következő példa egy 2 KB-os statikus RAM memóriablokkot ábrázol, $1K \times 4$ bites áramkörökből. Az adatbusz 16 bites, a címbusz pedig 20.



Megfigyelhető, hogy 1 KB-nyi memória két áramkörből áll, amelyek párhuzamosan dolgoznak, csak az adatvonalaik vannak úgy az adatbuszhoz kötve, hogy kijöjjön a 8 vezetékes csoport. A blokk által elfoglalt memóriatartomány $FC000_H-FC7FF_H$, ami csak egy szelekciós jel generálását teszi szükségessé, amint az a címhatárok táblázatából is látszik:

A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	...	A1	A0	Szelekció	Címterület
1	1	1	1	1	1	0	0	0	0	...	0	0	SEL=1	$FC000_H-FC7FF_H$
1	1	1	1	1	1	0	0	0	1	...	1	1		

A memória-áramkörök 10 címbemenettel rendelkeznek, ezekre az A1–A10 címvezetékeket vezetjük. A SEL szelekciójel egyenletét könnyen felírhatjuk a táblázat alapján:

$$SEL = A19 \cdot A18 \cdot A17 \cdot A16 \cdot A15 \cdot A14 \cdot \overline{A13} \cdot \overline{A12} \cdot \overline{A11} =$$

$$= A19 \cdot A18 \cdot A17 \cdot A16 \cdot A15 \cdot A14 \cdot (\overline{A13} + \overline{A12} + \overline{A11})$$

A bájt kiválasztást is figyelembe véve megkapjuk a csip kiválasztó jelek egyenletét:

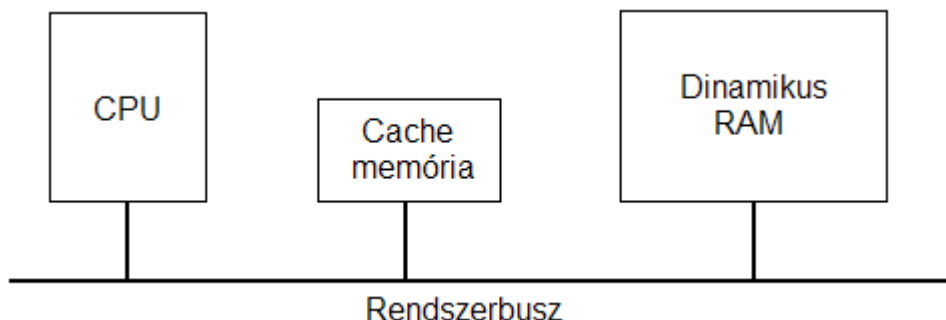
$$CS_{LOW} = SEL \cdot BE0$$

$$CS_{HIGH} = SEL \cdot BE1$$

A többi vezérlőjel (olvasás, írás) alkalmazása könnyen megérthető az általános elv alapján.

5. Cache memória

Egy **gyorsítótár** (*cache*) tartalmazó tárhierarchia blokkvázlatát mutatja az alábbi ábra. A processzor a buszon kiküldött címről kezdeményezi a memória-hozzáférést. Ha az adat benne van a cache-ben (*cache hit*), akkor ezt gyorsan megkapja, és folytathatja a feldolgozást. Ha az adat nincs a cache-ben (*cache miss*), azt a dinamikus RAM-ból kell kiolvasni, ami hosszabb ideig tart.

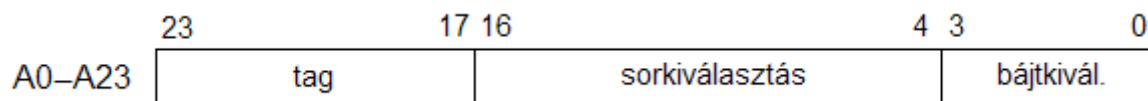


Példaként egy 128 KB-os, közvetlen leképezésű, átírási gyorsítótárát tárgyalunk. A közvetlen leképezés azt jelenti, hogy egy, a cache-be bemásolt adat mindig ugyanarra a helyre kerül. Az átírási (*write-through*) működés esetében az összes CPU-írás közvetlenül a főtárba kerül, de ha a hivatkozott cím megtalálható a cache-ben is, akkor az adataktualizálás ott is megtörténik. Minden sor 16 bájtos, így 32 bites adatbusz esetében egy sor feltöltéséhez 4 szó átvitelére van szükség (ezt *burst* átvitelben szokás megvalósítani). Kiszámítható, hogy a cache $128\text{ K} / 16 = 8\text{ K} = 8192$ sort tartalmaz. A cache vázlatos szervezése a következő:

tag 0	0. sor adat
tag 1	1. sor adat
tag 2	2. sor adat
⋮	⋮
tag 8191	8191. sor adat

7 bit16 bájt

Feltételezve, hogy a címbusz 24 bites (A0–A23), a címszó szerkezete a következőképpen alakul:

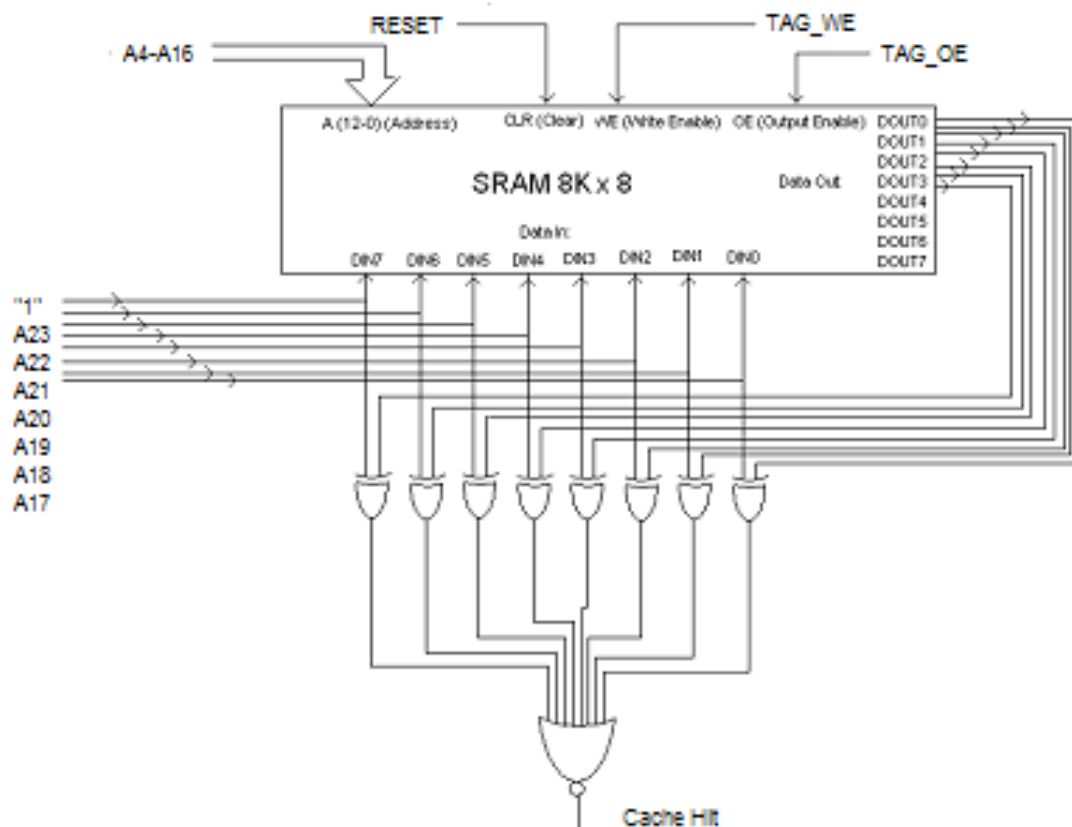


A legkisebb helyiértékű 4 bit azonosítja a bájtot egy soron belül, a következő 13 bit a sort választja ki, míg a fennmaradt 7 bit beazonosítja a keresett sort a lehetséges 128 közül. A címnek ez utóbbi része kerül összehasonlításra a kiválasztott sorban lévő taggal, és egyezőség esetén cache-hitről beszélünk, azaz a keresett adat a cache-ben van.

A gyorsítótár „tag” részének implementálásához két elemre van szükségünk:

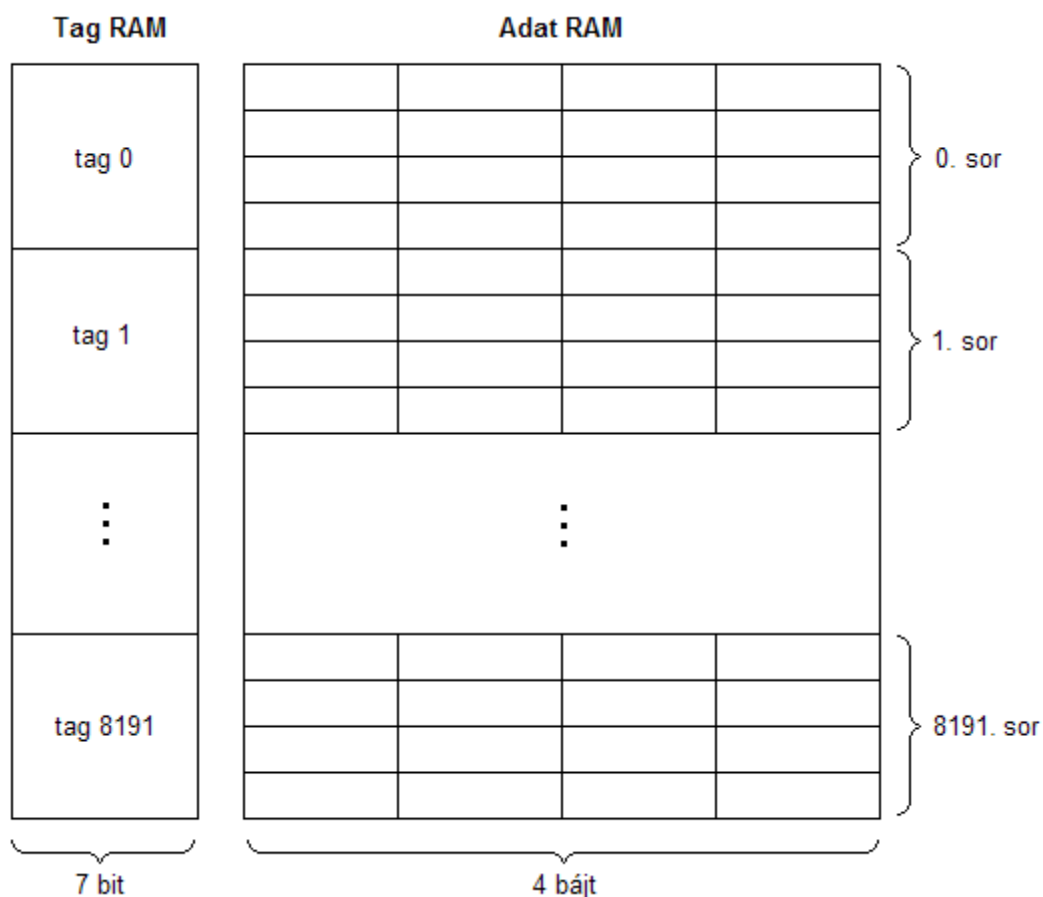
- egy $8\text{ K} \times 8$ bites statikus RAM memóriára
- egy 8 bites összehasonlító áramkörre

A „tag RAM” kapcsolási rajza a következő lesz:

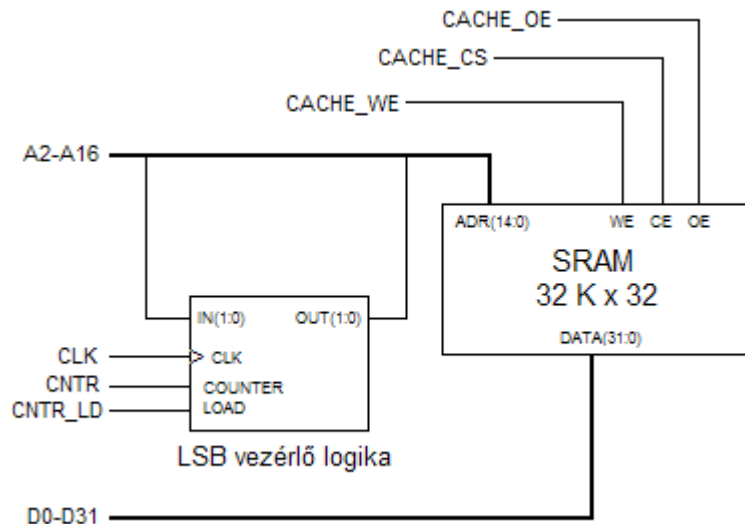


A tag memória adatbemeneteire az A17-A23 címvonalak kerülnek, amelyek egy sor beírásánál a tag részt tartalmazzák. Egy CPU-hivatkozásnál, a memória-áramkör címbemenetein azonosított rekeszben lévő adat kerül bitenként összehasonlításra (XOR kapusor) a processzor által hivatkozott cím magas helyiértékű bitjeivel. Egyezés esetén a keresett taggal rendelkező sor benne van a cache-ben, ekkor a *Cache Hit* jel aktiválódik. Figyeljük meg, hogy a DIN7 bemenet logikai „1”-re van kötve, ezáltal lehet megkülönböztetni az érvényes adatokat tartalmazó rekeszeket az üresektől (mivel RESET után ez egész áramkör tartalma törlődik). Az írásnál és olvasásnál használt vezérlőjeleket a cache memória-rendszer vezérlő logikája állítja elő.

A rendszer „adat” részének minden sora – amint említettük – 16 bájtot tartalmaz. Mivel az adatbusz 32 bites, az adatmemóriát is ilyen méretű egységekbe kell szervezni, ami azt jelenti, hogy az **„adat RAM”** 32 K × 32 bites lesz (mert a meghatározásban megadott 128 K × 8 bit = 32 K × 32 bit). Ez azt jelenti, hogy minden tagnek 4 darab 32 bites rekesz felel meg az adatrészben:



Az előző gondolatmenetnek megfelelően, az adatészt implementáló statikus RAM áramkör bekötési sémája a következő lesz:



A memóriablokk adatvonalai az adatbuszhoz csatlakoznak, ezen az útvonalon történik a cache feltöltése és kiürítése. A címszerkezetből láttuk, hogy egy-egy cache-sor azonosításához 13 címbitet használunk (A4–A16), mégis az adat RAM címbemeneteire 15 bites cím került (A2–A16). Ez azért van, mert 2 bit (A2 és A3) szükséges beazonosítani a soron belül, hogy melyik 32 bites rekeszben található a keresett egy bájtos adat. Ebből kifolyólag, a címbemenetek alsó két bitjének a kezelésére szükség van az ábrán látható LSB vezérlő logikára, amely egy multiplexert és egy számlálót tartalmaz. Cache hit esetén ezt a két bitet a multiplexer szolgáltatja a CPU-tól érkező címből (A2 és A3). Akkor viszont, amikor cache hitet követően egy sor feltöltése történik a dinamikus RAM-ból, a két címbit a számlálótól származik, amely négy rekeszen át inkrementálja a címet, a négy 32 bites szó feltöltésének ütemében.

A cache megfelelően időzített vezérlőjeleit egy, a gyorsítótárhoz tartozó vezérlő logika állítja elő. Ezt a lehetséges négy esetben (*read-hit*, *read-miss*, *write-hit*, *write-miss*) szükséges idődiagramok alapján, mint véges állapotú automatát lehet implementálni.

KÉRDÉSEK, TÉTELEK

- 5.1. Mekkora a központi memória címtartománya?
- 5.2. Mi a memória-áramkörök lineáris szelekciójának elve?
- 5.3. Mit értünk memóriatérkép alatt?
- 5.4. Mi a memória-áramkörök dekódolt szelekciójának elve?
- 5.5. Válaszolja fel és magyarázza el egy ROM memória-áramkör belső felépítését.
- 5.6. Adja meg a ROM memóriacsip buszrendszerhez csatlakoztatásának elvi vázlatát.
- 5.7. Jelölje meg egy memória-áramkör jellemző idődiagramján a hozzáférési időt.
- 5.8. Adja meg a statikus RAM memóriacsip buszrendszerhez csatlakoztatásának elvi vázlatát.

5.9. Mi célt szolgálnak a bájt kiválasztó jelek?

5.10. Milyen két tárolórészből áll egy cache-rendszer? E tárolórészek címezéséhez a címszó mely mezőit kell használni?

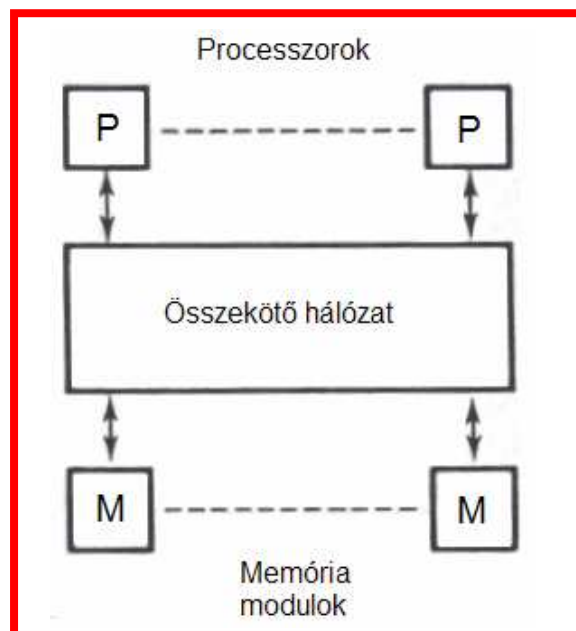
5.11. Mit jelent, és hogy állapítható meg a cache-hit?

7. MULTIPROCESSZOROS RENDSZEREK

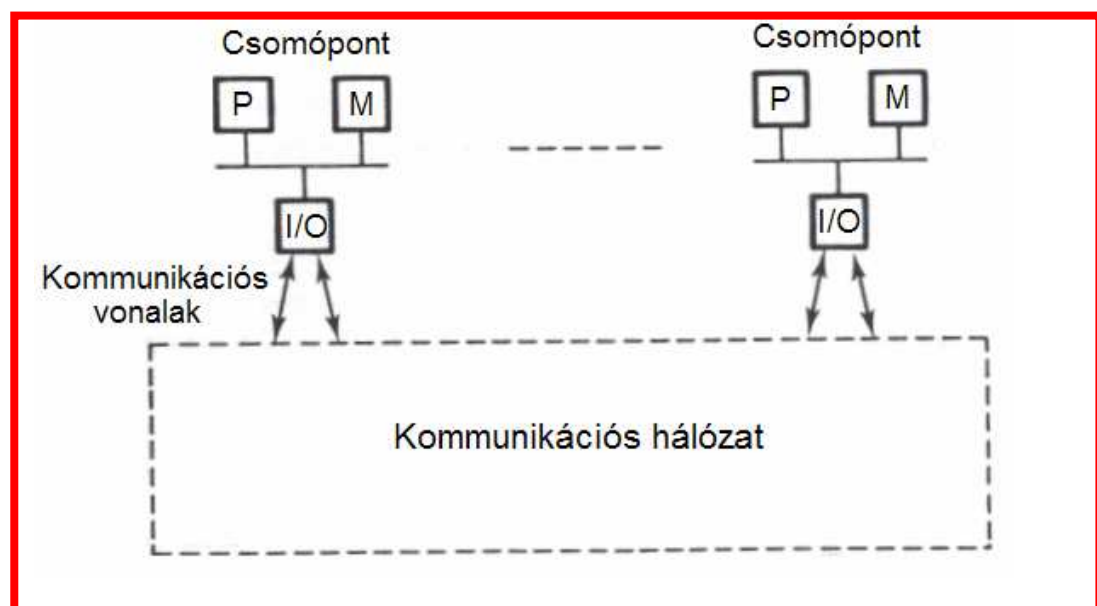
7.1. A multiprocesszoros rendszerek osztályozása

A multiprocesszoros rendszerek jellemzője, hogy egyidejűleg több független processzor dolgozza fel az alkalmazás adatait. Mindegyik processzor önálló programot hajt végre, de az alkalmazás megkívánja, hogy a folyamatok adatokat adjanak át egymásnak. Az adatátadás megvalósítása szempontjából, a multiprocesszoros rendszerek **két nagy csoportját** különböztetjük meg:

- **Megosztott memóriájú** (*shared memory*) rendszerek – a processzorok egy összekötő hálózaton (*interconnection network*) keresztül kapcsolódnak egy vagy több közösen használt memóriamodulhoz, amelyeken keresztül adatokat cserélhetnek:



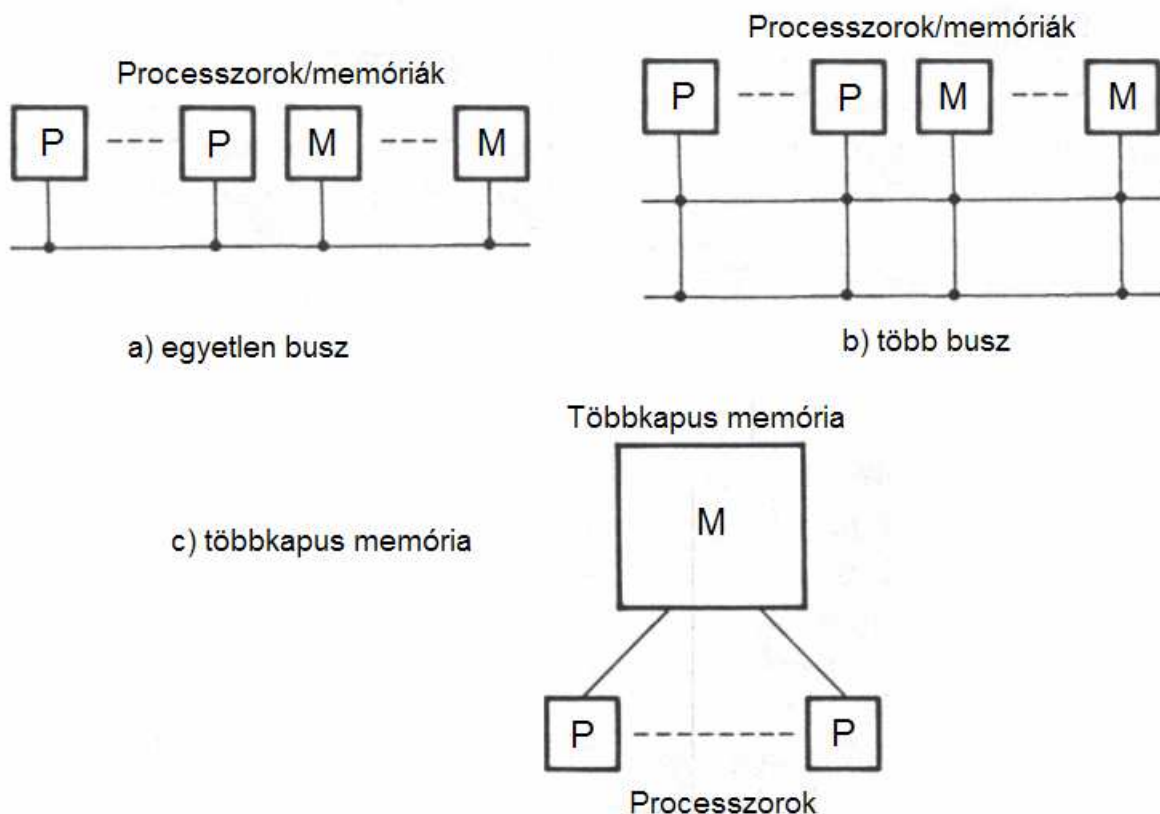
- **Elosztott** (*distributed*) vagy **üzenetátadásos** (*message passing*) rendszerek – mindegyik processzor rendelkezik csak saját memóriával, és üzenetek formájában cserél adatokat a többi processzorral, általában közvetlen kommunikációs vonalakon:



1. A megosztott memóriájú rendszereket két szempont szerint szokás osztályozni:

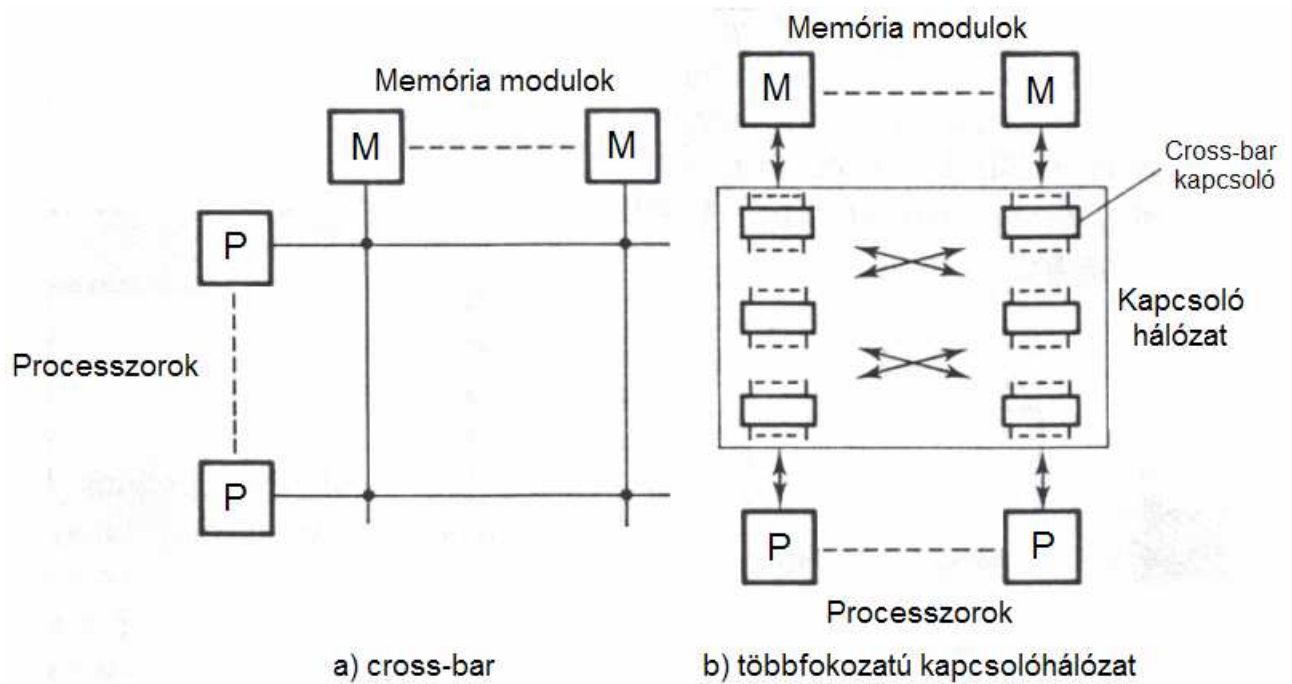
a) Az összekötő hálózat struktúrájának jellege alapján két típust különböztetünk meg:

- **Statikus** összeköttetésű rendszereket – amikor a hálózat nem tartalmaz kapcsolóelemeket. Az ilyen rendszer lehet
 - egy buszos (*single bus*);
 - több buszos (*multiple bus*);
 - többkapus memóriával (*multiport memory*) rendelkező.



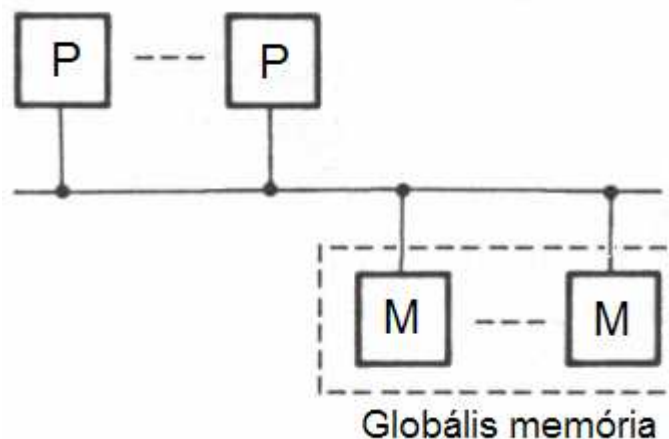
• **Dinamikus** hálózatú rendszerek – amikor a hálózat kapcsolóelemeket tartalmaz. Ez lehet

- egy fokozatú kapcsolóhálózat, ami a *cross-bar*;
- több fokozatú kapcsolóhálózat (*multistage interconnection network*).

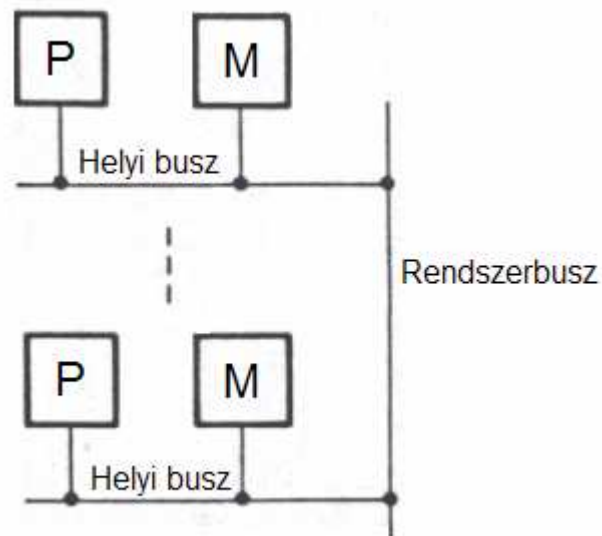


b) A megosztott memóriához való hozzáférés szerint léteznek:

- **UMA** (*Uniform Memory Access*) rendszerek – amelyeknél a közös memóriaterülethez való hozzáférés ideje azonos minden processzor számára (ezeket szokták még **szimmetrikus** multiprocesszoros rendszereknek is nevezni). Példaként megemlíthető az egyetlen globálisan elérhető memóriával rendelkező rendszer:

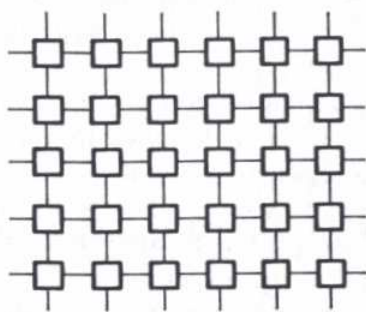


- **NUMA** (*Non-Uniform Memory Access*) rendszerek – amelyeknél a közös memóriához való hozzáférési idő eltér a processzorok között, attól függően, hogy egy bizonyos processzor a megosztott memória mely területére hivatkozik. Például ilyen rendszert kapunk, ha mindegyik processzor rendelkezik lokális memóriával, amelyet viszont megoszt a többi processzorral. Ekkor a hozzáférési idő a saját memóriájához rövidebb lesz, mint a többi memóriához:

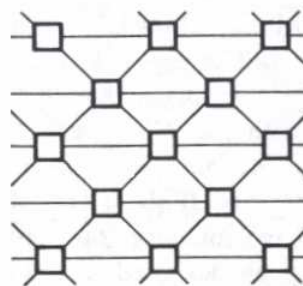


2. Az **elosztott (üzenetátadásos)** rendszereket a következőképpen osztályozhatjuk:

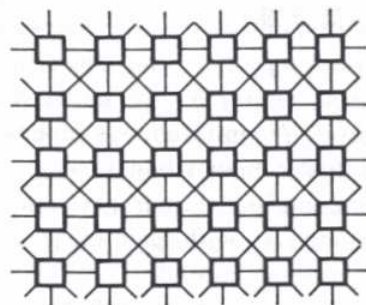
- **Erősen párhuzamos processzorok** (MPP – *Massively Parallel Processors*) – amelyek szabályosan rögzített összekötési hálózattal rendelkeznek, mint például
 - négy-, hat-, nyolckapcsolatú háló (a négykapcsolatúra példa a *transzputer*)
 - teljesen összefüggő hálózat
 - kocka, hiperkocka
 - fa struktúra



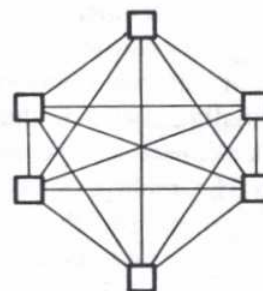
a) négykapcsolatú háló



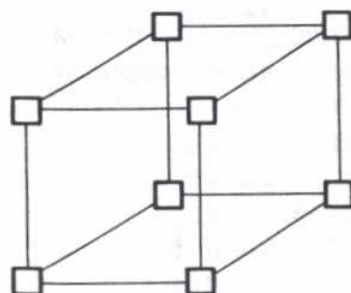
b) hatkapcsolatú háló



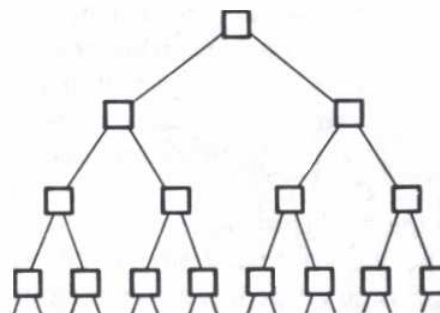
c) nyolckapcsolatú háló



d) teljesen összefüggő hálózat



e) kocka

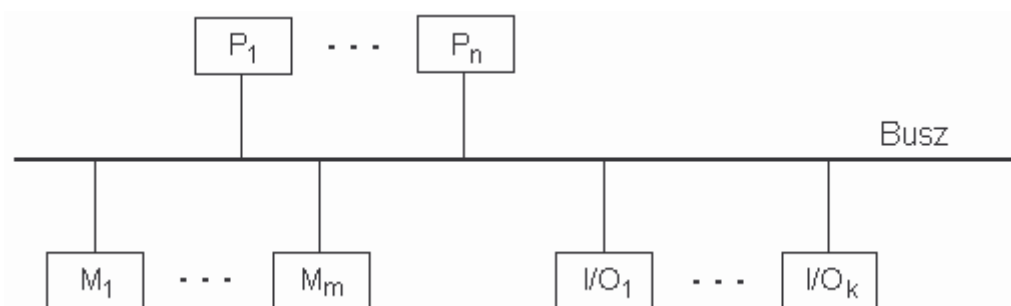


f) fa

- **Klaszterek** – amelyek egymással egy hálózaton (általában helyi hálózaton, LAN-on) keresztül összekapcsolt teljes számítógépek, és együttműködésükkel egyetlen erőforrás illúzióját keltik. Ilyen például egy munkaállomásokból álló COW (*Cluster of Workstations*). Magas rendelkezésre állásuk, skálázhatóságuk és teljesítőképességük miatt szerverek kialakítására is alkalmasak.

7.2. Egybuszos rendszerek jellemzői

Ebben az esetben a rendszer moduljai (processzor, memória, I/O) egyetlen buszhoz csatlakoznak:



A memória és I/O modulok minden processzor számára elérhetők, ezeket megosztottan használhatják. (Ez nem zárja ki, hogy a processzormoduloknak legyenek saját memória és I/O erőforrásai is.) A modulok két csoportba sorolhatók:

- **Master** (mester) modulok – amelyek képesek egy adatátvitelt vezérelni a buszon, miután átvették ennek használatát. Ilyenek a processzor- és a DMA-modulok.
- **Slave** (szolga) modulok – amelyek passzív elemekként vannak a buszra kötve, és várják, hogy egy master kezdeményezzen velük adatátvitelt, amelyre válaszolnak. A slave modulok tipikus példái a memória- és az I/O modulok.

Mivel a buszt egy adott pillanatban csak egy master használhatja, a buszidőt meg kell osztani a mesterek között (*time sharing*). Annak eldöntésére, hogy az igénylők közül ki kapja meg a buszhasználat jogát, valamilyen arbitrációs mechanizmust használnak.

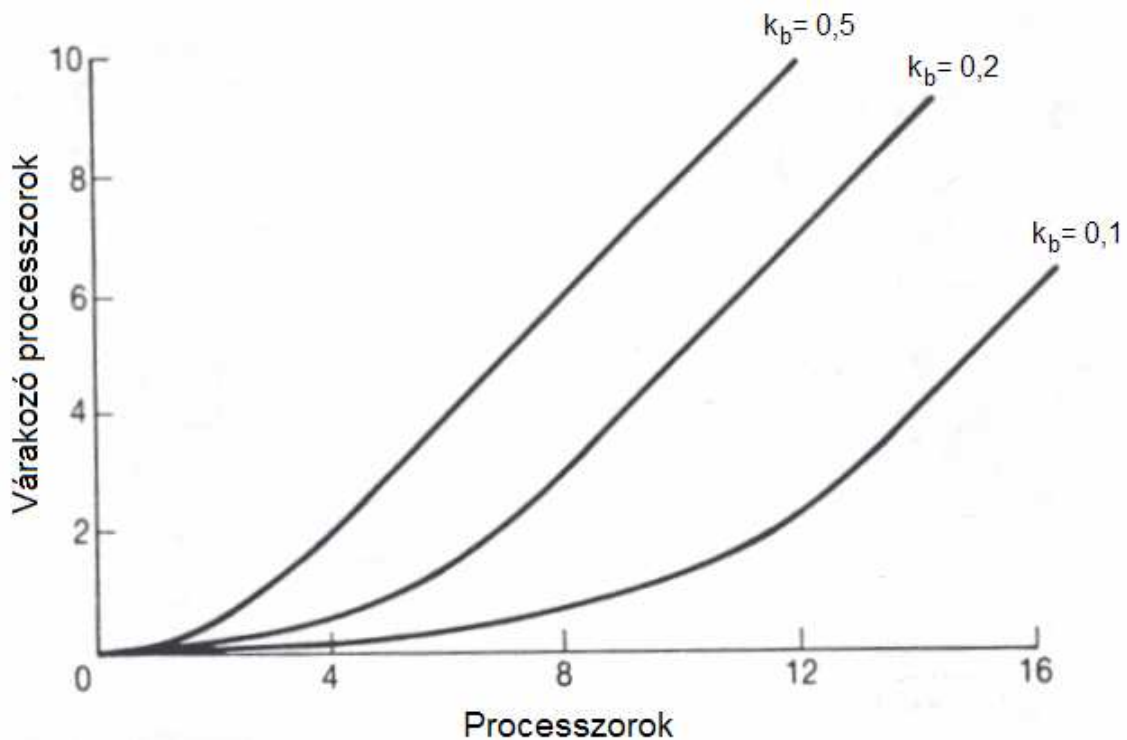
Az egybuszos multiprocesszoros rendszer tervezése és implementálása egyszerű és nem költséges feladat. Könnyen konfigurálható az igényeknek megfelelő rendszer, amely a későbbiekben továbbfejleszthető. A legnagyobb gondot az egyetlen adatútvonal – a busz – magas forgalma jelenti, ami a rendszerbe kapcsolható processzor modulok számát korlátozza. Ha mindegyik processzor állandóan igényelné a buszt, akkor semmilyen hatékonysága sem volna a több processzor alkalmazásának, mivel csak az egyik használná a buszt, a többi pedig várakozna. Ha statisztikailag feltételezzük, hogy a processzorok egy hosszabb T időszakból egyformán t_b ideig igénylik a buszt, akkor elvileg mindegyiknek biztosított a hozzáférése mindaddig, amíg $n t_b \leq T$, ahol n a processzorok száma. A $k_b = \frac{t_b}{T}$

arányt buszhozzáférési együtthatónak nevezzük – értéke egy-egy processzornál annál kisebb, minél nagyobb a program által diktált autonómiája. Akkor, amikor egy újabb processzort hozzáadva a rendszerhez, az egyik processzornak már nem jut hozzáférési idő a buszhoz (várakoznia kell), elértük a **busz telítettségét** (*bus saturation*). A telítettségi küszöbnek megfelelő processzorszám $n_s = \frac{1}{k_b}$. A

várakozásra kényszerülő processzorok számát (n_w) a következőképpen számíthatjuk ki:

$$n_w = \begin{cases} 0 & \text{ha } n \leq \frac{1}{k_b} \\ n - \frac{1}{k_b} & \text{ha } n > \frac{1}{k_b} \end{cases}$$

Ennek a függvénynek a grafikonját – különböző k_b értékek számára – az alábbi ábra mutatja, azzal a megjegyzéssel, hogy a valós görbék nem törtvonalak, mert a gyakorlatban a telítettség elérése előtt is vannak várakozó processzorok, ha többen igénylik egyszerre a buszt:

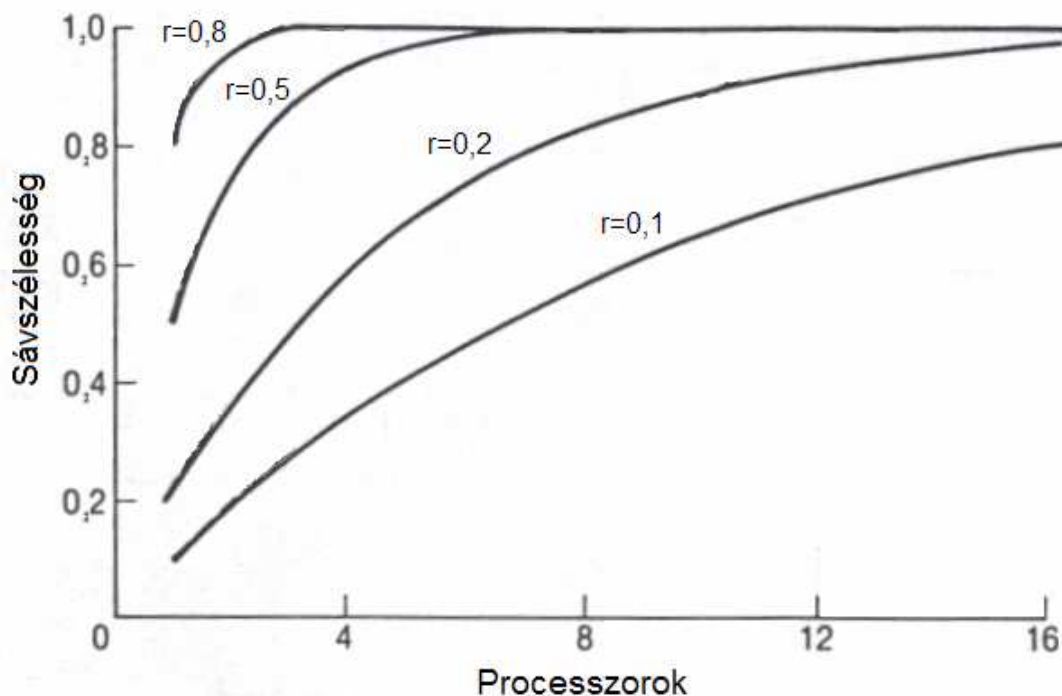


Az egybuszos rendszer tervezésénél figyelembe kell venni, hogy a buszhoz csatlakoztatott processzorok száma ne vezessen telítettséghez. Telítettség után ugyanis újabb processzorok csatlakoztatása már nem növeli a rendszer hatékonyságát, nem vezet további párhuzamos feldolgozáshoz, csak növeli a várakozó processzorok számát.

Könnyen kiszámíthatjuk a **busz sávszélességét** (*bus bandwidth*) is, azaz a buszt átlagosan használó processzorszámot. Feltételezzük, hogy a processzorok véletlenszerűen generálnak hozzáférési kérést a megosztott memóriához, és egy processzor által kezdeményezett kérés valószínűsége r (hozzáférési ráta). Annak a valószínűsége, hogy a processzor ne kérjen hozzáférést $1-r$, annak pedig, hogy egy processzor se kérjen $(1-r)^n$. Továbbá, annak a valószínűsége, hogy legyen legalább egy kérés $1-(1-r)^n$. Ez az érték éppen a busz átlagos foglaltságát, azaz ennek a sávszélességét jelenti:

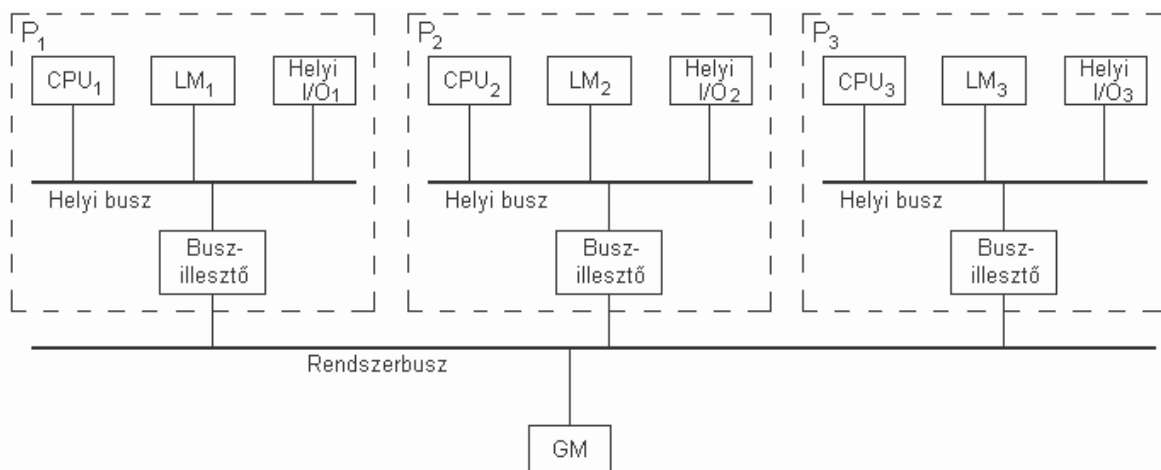
$$BW = 1 - (1 - r)^n$$

A sávszélesség grafikonját az alábbi ábra tartalmazza. Látható, hogy minél nagyobb a hozzáférési ráta, annál hamarabb telítődik a busz.

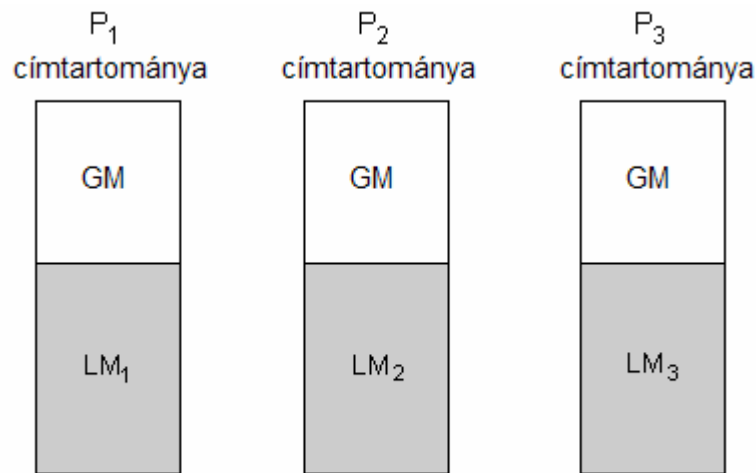


7.3. Egybuszos rendszerek címkiosztása

A buszhoz csatlakoztatott összes memóriamodul egyetlen globális, megosztott memóriát (GM) alkot, amelyhez mindegyik processzor hozzáférhet. A globális memóriát célszerű a közös adatok tárolására használni. Ezen kívül, minden egyes processzor rendelkezik egy saját, lokális memóriával (LM), amelyet a saját, helyi sínén érhet el. A két sín közötti kapcsolatot egy buszillesztő egység biztosítja:



Két processzor közötti adatátvitel két lépésben zajlik, két hozzáféréssel a rendszerbuszhoz: a küldő processzor beírja az adatot a globális memóriába, majd a célprocesszor beolvassa onnan egy újabb buszhozzáféréssel („postafiók” elv). Ahhoz, hogy mindkét processzor hozzáférhessen a közös memóriához, ennek ugyanazt a területet kell elfoglalnia mindkét processzor címtartományában. A címtartomány fennmaradó területe a processzorok lokális memóriája számára marad meg. Például, egy három processzormodult tartalmazó rendszer esetében, a processzorok címtartományát az alábbi ábra szerint lehet kialakítani:



7.4. A buszhozzáférés

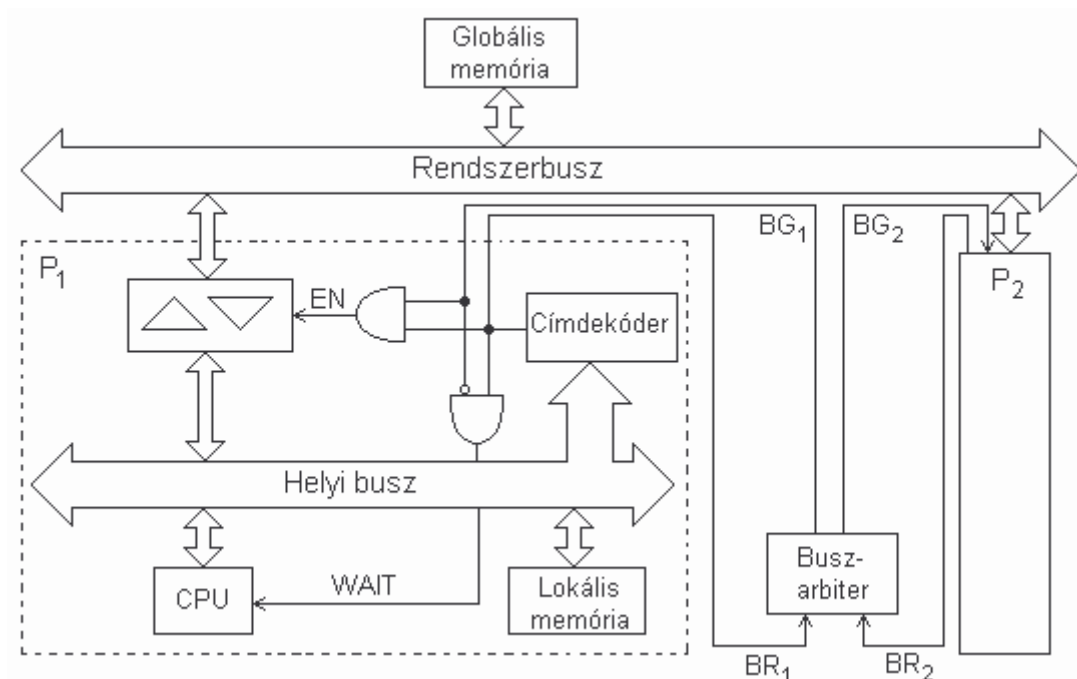
A processzor helyi sínje és a rendszerbusz között a kapcsolatot a **buszillesztő egység** biztosítja, ami a buszorientált multiprocesszoros rendszer tervezésének sajátos eleme. A buszillesztő egység fő feladata lehetővé tenni az illető processzor hozzáférését a rendszerbuszhoz egy adatátvitel (írás, olvasás) végrehajtása céljából a globális memóriával. A processzor akkor fog buszhozzáférést igényelni az illesztőtől, ha a programja végrehajtása során egy olyan adatátviteli utasításhoz ér, amelynek forrása vagy célja a globális memóriában van.

A buszillesztő egység két fő elemet tartalmaz:

- **Vezérlő logikát** – amely figyeli és dekódolja a helyi buszon lévő címet, hogy eldöntse, mikor vonatkozik a külső erőforrásra
- **Puffereket** – amelyek a két busz azonos vonalai közötti jelátvitelt biztosítják. Hozzáférés hiányában a pufferek kimenetei magas impedanciájú állapotban vannak.

Buszhozzáférési igény esetén, ha minden feltétel teljesült (a busz szabad és nincs más, magasabb prioritású igénylő) a vezérlő logika kiadja a hozzáférést visszaigazoló jelet, ami megnyitja a puffereket és így az átvitel megvalósulhat.

A buszhozzáférés egyszerűsített vázlatát mutatja a következő ábra:

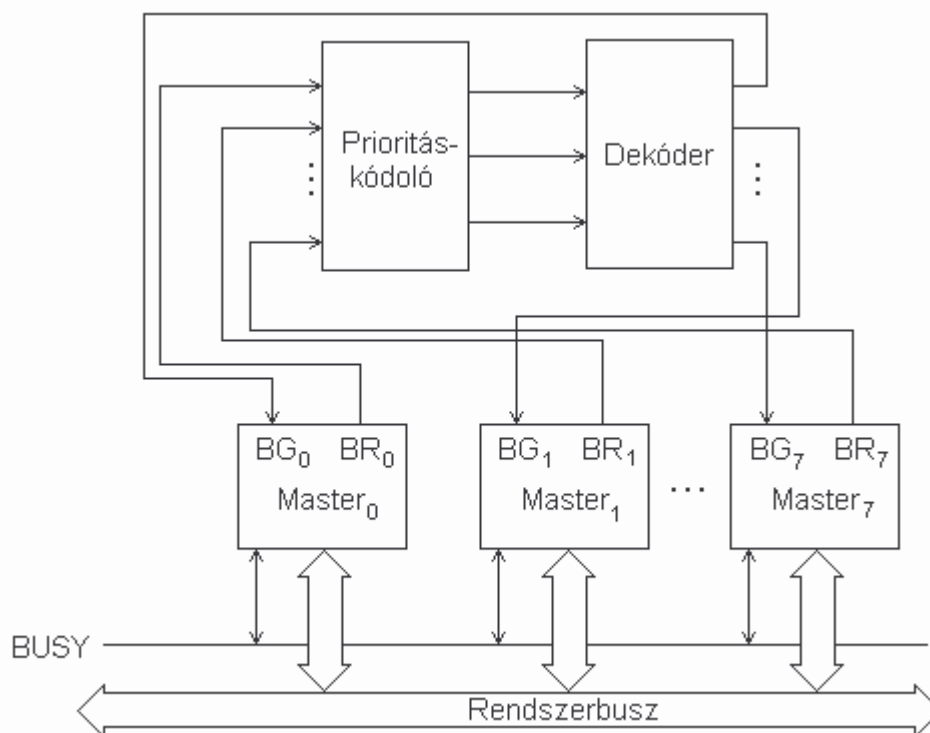


Amikor a dekóder egy, a megosztott memória-tartományba tartozó címet ismer fel, akkor egy buszkérélmét generál (BR – *Bus Request*) a buszarbiter felé. Ameddig a hozzáférési jogosultság nem valósul meg és a busz nem szabadul fel, a processzort várakozó állapotba kell kényszeríteni a WAIT jellel. Egyidejű igények esetén a buszarbiter a legnagyobb prioritással rendelkező processzormodul felé fog visszaigazoló jelet (BG – *Bus Grant*) küldeni. Ennek a jelnek az aktiválása után megtörténik a pufferek megnyitása (EN=1) és a processzor kikerül a várakozó állapotból, befejezi a buszciklust.

A buszarbitráció megvalósítására két módszert alkalmazhatunk:

- **Párhuzamos arbitráció** – amikor minden master egyszerre történik az igények elbírálása valamilyen prioritási algoritmus szerint (fix, forgó, LRU stb.);
- **Soros arbitráció** – amikor a masterek láncba vannak kapcsolva a prioritásuk csökkenő sorrendjében, és ebben a sorrendben történik az igények feldolgozása.

A párhuzamos arbitrációra mutat példát a következő ábra:

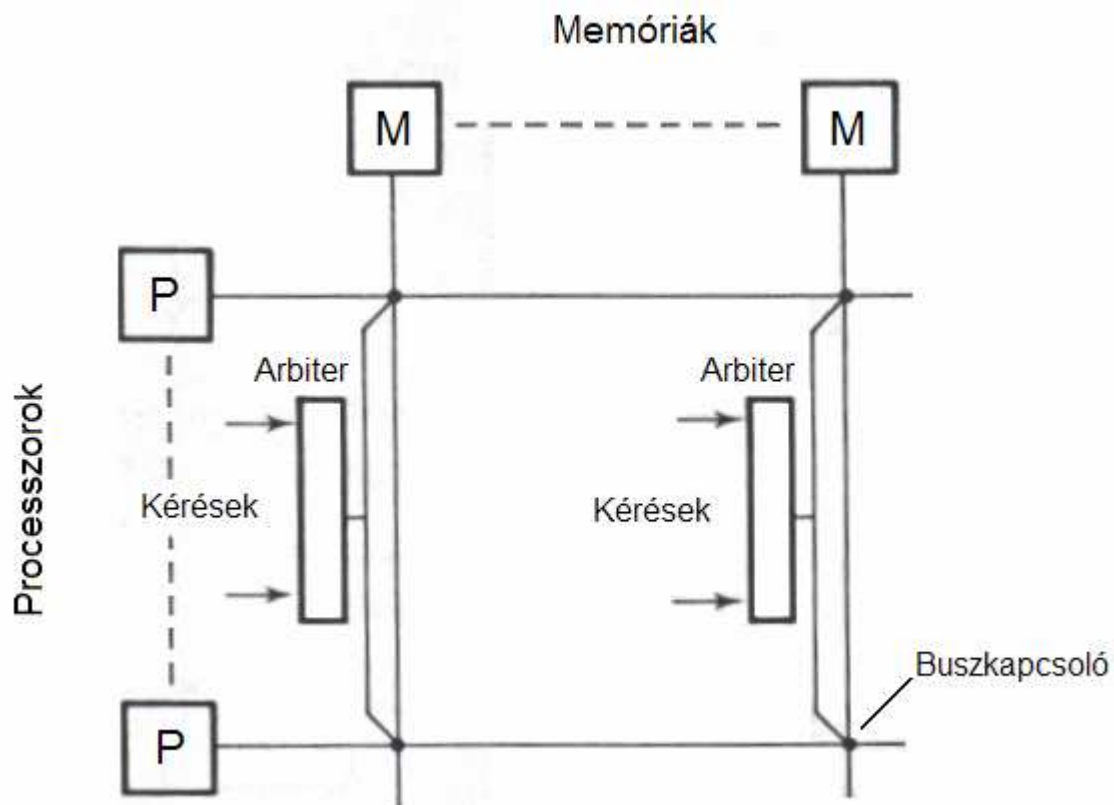


A buszarbitert egy prioritáskódoló és egy dekódoló áramkör valósítja meg. A prioritáskódoló bemenetei rögzített, csökkenő sorrendben lévő prioritással rendelkeznek, ide érkeznek a buszigények (BR). A kimeneteken az aktív bemenetek közül a legmagasabb prioritással rendelkezőnek a sorszáma jelenik meg. Ez a sorszám kiválasztja a dekóder egyik kimenetén azt a buszhozzáférést engedélyező jelet (BG), amelyik a legmagasabb prioritással rendelkező masterhez tartozik az igénylők közül. A hozzáférés másik feltétele, hogy az előző átvitelt végrehajtó master szabadítsa fel a buszt (BUSY=0).

7.5. A cross-bar rendszer

Ebben a rendszerben a processzorok és a memóriák egy mátrix alakú kapcsolóhálózattal vannak összekötve úgy, hogy mindegyik processzor-memória összeköttetésre jut egy kapcsoló. Így egyszerre minden processzor dolgozhat, természetesen külön-külön memóriával – ezért az ilyen fajta rendszert *blokkolásmentesnek* nevezik.

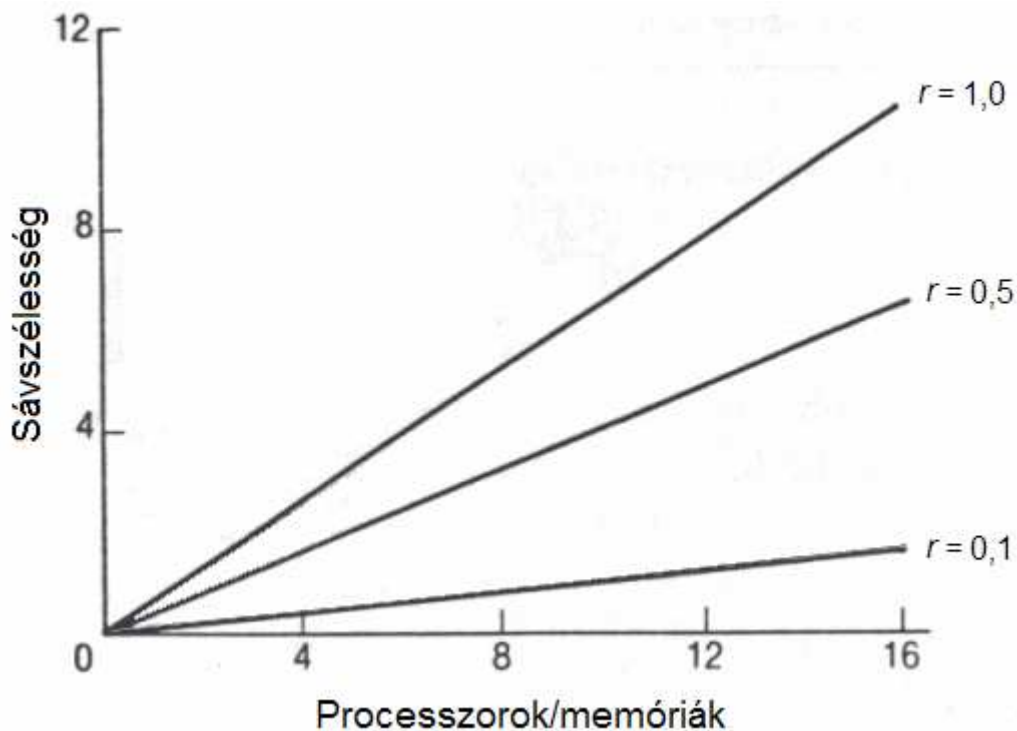
Az elemek közötti összekötő vonalak valóban buszokat jelentenek, így egy-egy kapcsolónak 50-100 jel továbbítását kell kezelnie. Minden memóriamodulhoz tartozó oszlopnak van egy saját arbitrációs logikája, amely eldönti, hogy melyik processzor jogosult az illető memóriához való hozzáférésre. A döntésnek megfelelően fog „bezárni” az igénylők közül legnagyobb prioritással rendelkező processzor sorában lévő kapcsoló. A processzorok memória-hozzáférési igényei az illető memóriamodulhoz tartozó arbiterhez futnak be:



A **cross-bar rendszer sávszélességének** kiszámításához legyen r annak a valószínűsége, hogy egy processzor kérést intéz a megosztott memóriához. Ha m a memóriamodulok száma, akkor annak a valószínűsége, hogy ez a kérés egy bizonyos modulra vonatkozzon r/m . Annak a valószínűsége, hogy a kérés nem egy bizonyos memóriamodulra vonatkozik $1-r/m$. Ha p a processzorok száma, akkor annak a valószínűsége, hogy egy processzor sem hivatkozik erre a bizonyos memóriára $(1-r/m)^p$. Ebből következik, hogy $1-(1-r/m)^p$ lesz annak a valószínűsége, hogy legalább egy processzor hivatkozik erre az egy memóriamodulra. Most már kiszámítható a cross-bar rendszer sávszélessége, azaz az aktív (hivatkozott, tehát adatátvitelt lebonyolító) memóriamodulok száma:

$$BW = m \left(1 - \left(1 - \frac{r}{m} \right)^p \right)$$

A sávszélesség grafikonja az alábbi ábrán látható (abban az egyszerűbb esetben, amikor $p=m$). Megfigyelhető, hogy ha minden ciklusban történik memóriahivatkozás ($r=1$), akkor is maradnak várakozó processzorok (az egyenes nem 45 fokos), mert több processzor is hivatkozhat ugyanarra a memóriamodulra.



7.6. Többfokozatú kapcsolóhálózatok

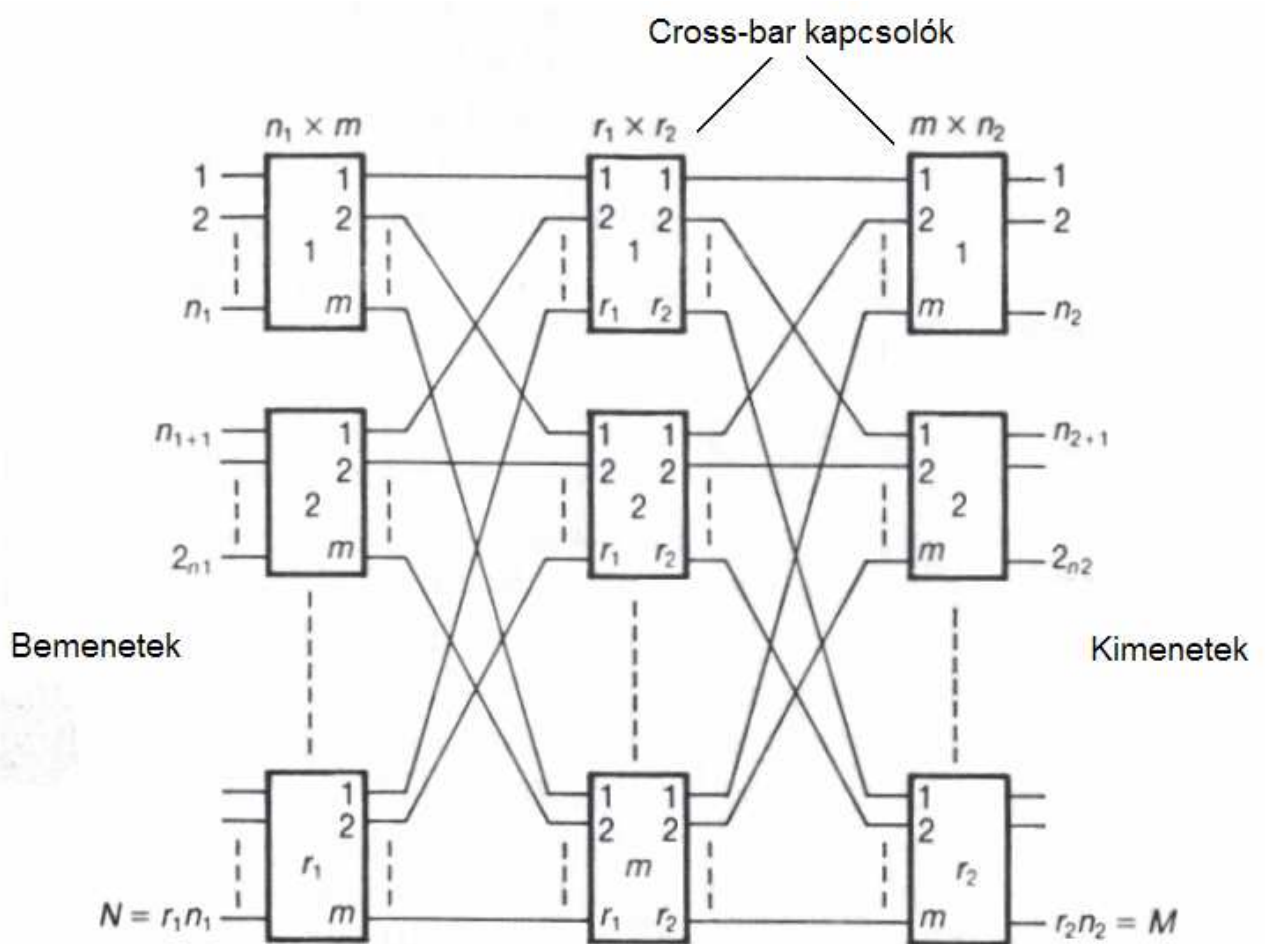
Ezekben a dinamikus hálózatokban (angolul *Multistage Interconnection Networks*) két csomópont (processzor, processzor/memória) közötti összeköttetést több szinten elhelyezett kapcsolók biztosítják, amelyeket különböző kapcsolatok létrehozására lehet beállítani. A bemenettől a kimenetig haladó adatok több kapcsolófokozaton keresztül haladnak át. (Természetesen a kapcsolat lehet kétirányú is.) A hálózat lehet:

- **blokkolásmentes** – ha egyidejűleg minden egyes bemenet összekapcsolható egy-egy kimenettel, és ezt az összes bemenet/kimenet kombinációra meg lehet valósítani;
- **blokkoló** – ha nem teszik lehetővé az összes kombinációban az egyidejű bemenet/kimenet kapcsolatot.

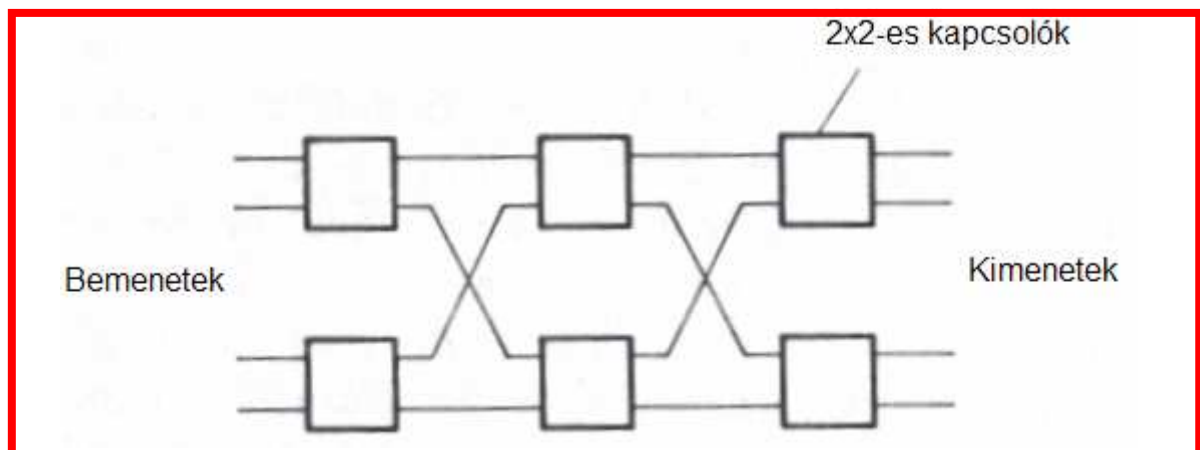
A többfokozatú kapcsolóhálózatnak két típusa van:

- Cross-bar kapcsolókkal ellátott hálózat
- Cella alapú hálózat (ez az előbbinek a sajátos esete, 2x2-es cross-bar kapcsolókkal)

Az alábbi ábra egy háromfokozatú, cross-bar kapcsolókkal kialakított **Clos hálózatot** mutat. A hálózat blokkolásmentes, ha $m \geq n_2 + n_1 - 1$.



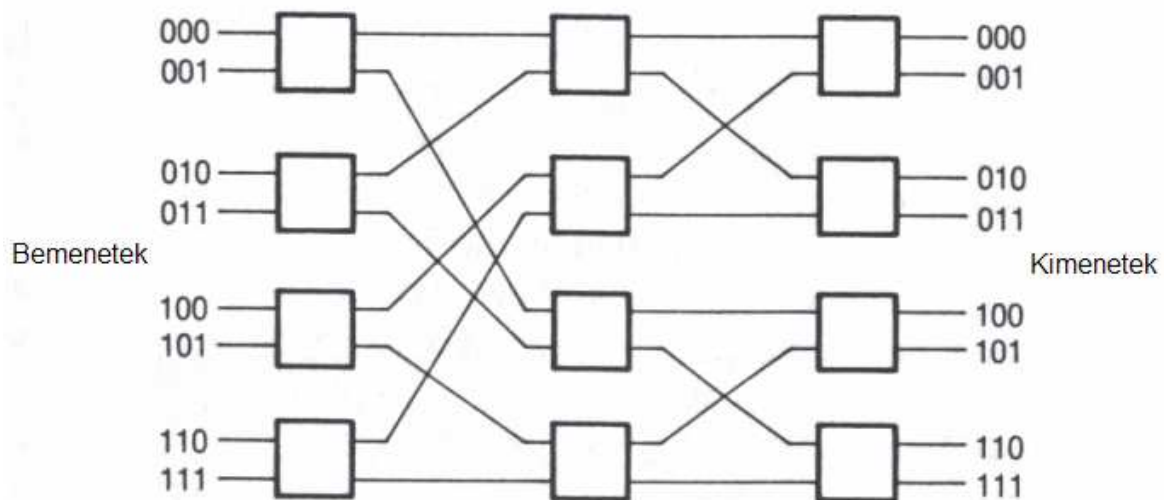
A Clos hálózat sajátos esete a két bemenetű és két kimenetű cella alapú **Benes hálózat**:



A cellák két állapotban lehetnek, párhuzamosan vagy keresztképpen kapcsolhatják össze a bemenetet a kimenettel:

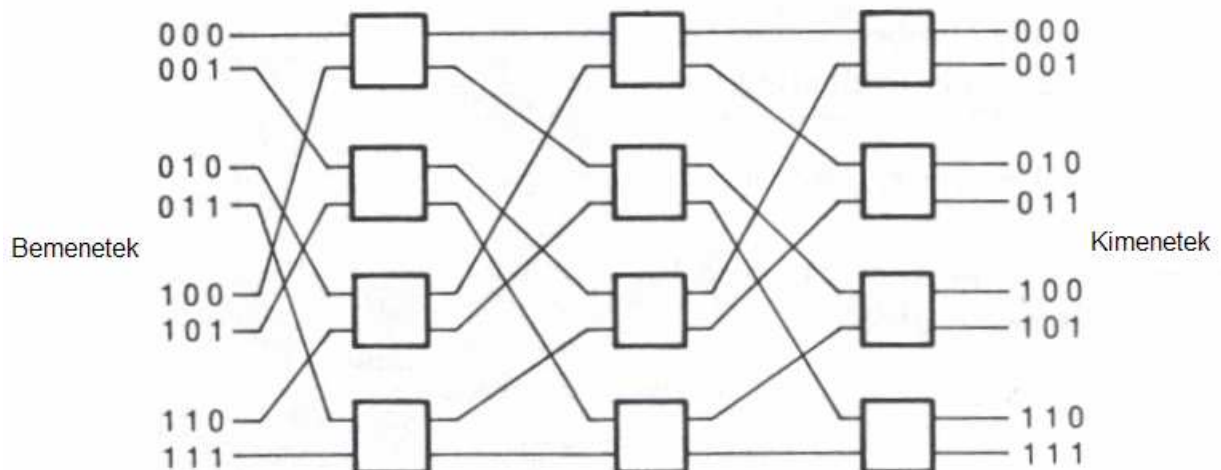


A cella alapú hálózatok általában blokkoló típusúak. Az alábbi ábrán bemutatott **baseline hálózat** is ilyen, de rendelkezik az **automatikus útválasztás** (*self-routing*) tulajdonságával: a célkimenet címének bitjei sorban a következő fokozat kapcsolóit irányítják (ha a bit=0, akkor párhuzamos kapcsolás, ha a bit=1, akkor keresztirányú kapcsolás). Ezekben a hálózatokban mindegyik bemenet és kimenet között csak egy útvonal lehetséges, ezért nem tolerálják a kapcsolók meghibásodását.



A baseline hálózat minden egyes fokozata megfelel a további lehetséges útvonalakat: az első fokozat után az adat a hálózat felső vagy az alsó felébe kerül, a második után valamelyik negyedbe, és végül a célba. Ha a bemenetek és kimenetek száma $N=2^n$, akkor a hálózat fokozatainak száma $\log_2 N$.

A többfokozatú kapcsolóhálózatok kialakításának egy másik változata a tökéletes keverésű hálózat (*perfect shuffle network*). Ebben az esetben egy fokozat bemeneteinek első fele közé beékelik a bemenetek második felét. Ilyen az **Omega hálózat**, amely blokkoló, de rendelkezik az önálló útvonalválasztás tulajdonságával:



A **többfokozatú kapcsolóhálózat sávszélességét** a kapcsolócellákkal megvalósított típusokra számítjuk ki. Egy kapcsolócella lényegében egy 2×2 -es cross-bar, tehát az előző alfejezetben meghatározott egyenleteket itt alkalmazhatjuk,

azzal a pontosítással, hogy $p=m=2$. Ha a bemeneti processzorok a kimenetre kötött memóriamodulokra r_0 valószínűséggel hivatkoznak, akkor:

- az első fokozati kapcsoló egyik kimenetén a memóiahivatkozás valószínűsége

$$r_1 = 1 - \left(1 - \frac{r_0}{2}\right)^2$$

- a második fokozati kapcsoló egyik kimenetén a memóiahivatkozás valószínűsége

$$r_2 = 1 - \left(1 - \frac{r_1}{2}\right)^2$$

.....

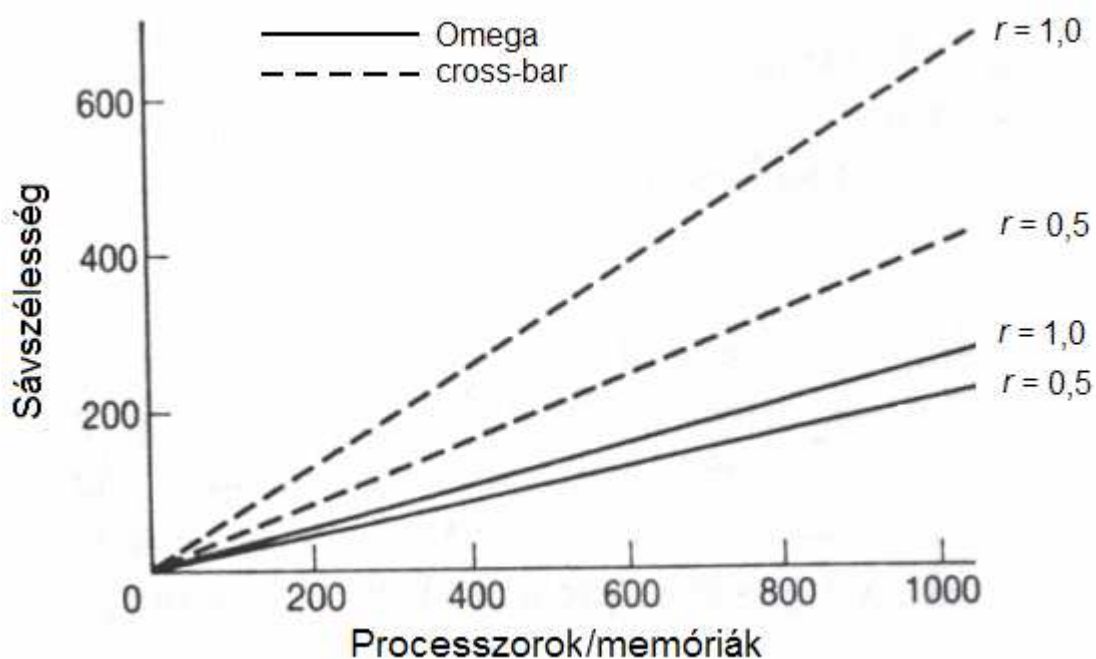
- az utolsó (n -edik) fokozati kapcsoló egyik kimenetén (azaz a hálózat egyik kimenetén) a memóiahivatkozás valószínűsége

$$r_n = 1 - \left(1 - \frac{r_{n-1}}{2}\right)^2$$

Most már kiszámítható a hálózat sávszélessége, vagyis hogy hány kimenetre (a 2^n -ből) történik hivatkozás:

$$BW = 2^n r_n$$

Az r_n kiszámítása rekurzívan történik, a fenti egyenletek alapján. Egy Omega hálózat sávszélességét a bemenetek számának függvényében mutatja az alábbi ábra. Összehasonlításként fel van tüntetve a hasonló számú bemenettel és kimenettel rendelkező csoss-bar hálózat sávszélessége is. Látható, hogy a többfokozatú hálózat sávszélessége kisebb, mert itt egyes memóriamodulokra való hivatkozás nem teljesíthető, a blokkolt útvonal miatt.



KÉRDÉSEK, TÉTELEK

- 7.1. Mit értünk megosztott memóriájú multiprocesszoros rendszeren? Vázolja fel az elvi felépítését, és sorolja fel a különböző típusait!
- 7.2. Mit értünk üzenetátadásos (elosztott) multiprocesszoros rendszeren? Vázolja fel az elvi felépítését, és sorolja fel a különböző típusait!
- 7.3. Vázolja fel az egybuszos multiprocesszoros rendszer felépítését! Mi a szerepük a master és slave moduloknak?
- 7.4. Mi a szerepe a buszillesztő egységnek? Milyen részegységeket tartalmaz, mi ezeknek a szerepe?
- 7.5. Magyarázza el a buszhozzáférés menetét, egy kisegítő ábrát is felhasználva!
- 7.6. Vázoljon fel egy cella alapú többfokozatú kapcsolóhálózatot! Mit jelent a blokkolásmentes hálózat? Hogy történik az automatikus útválasztás?

8. HIBATŰRŐ ARCHITEKTÚRÁK

8.1. A hibatűrés fogalma és alkalmazásai

A rendszerek meghibásodás nélküli működése olyan cél, amit sosem lehet elérni. Egyrészt a rendszer alkotóelemei hibásodhatnak meg, másrészt még egy tökéletesnek hitt terv is tartalmazhat felderítetlen hibákat. A magas megbízhatóság elérésének elvileg két lehetősége van: a hibák megelőzése (elkerülése a tervezés és implementálás közben, illetve eltávolítása a tesztelés során) és tolerálása (a működés közben).



A **hibatűrő rendszer** (*fault tolerant system*) képes ellátni a specifikált feladatait korlátozott számú hibás alegység (hardver és szoftver) jelenlétében is. A tolerálandó hibák lehetnek

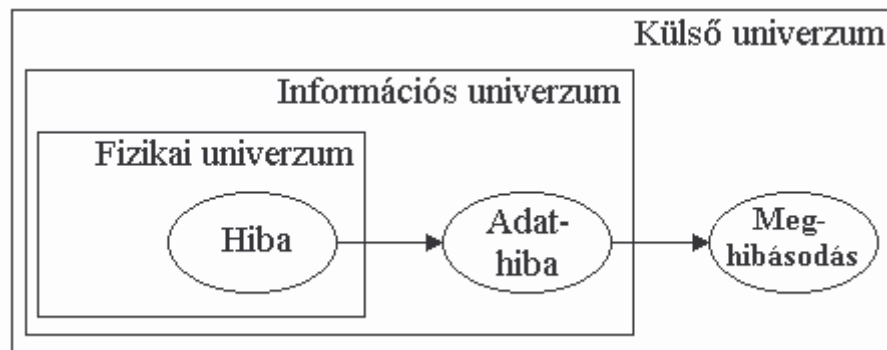
- Előre látott hibák (*anticipated faults*) – a meghibásodott hardver alkatrészekre vonatkoznak, ezekre lehet számítani
- Előre nem látott hibák (*unanticipated faults*) – a hardverben és szoftverben található tervezési hibák

Ha a rendszer működése nem a várakozásoknak megfelelően zajlik, akkor azt mondjuk, hogy a rendszer meghibásodott. A **meghibásodás** (*failure*) jelenti a rendszer viselkedésének első eltérését a specifikációtól. Ha már megtörtént, a meghibásodást nem lehet visszafordítani.

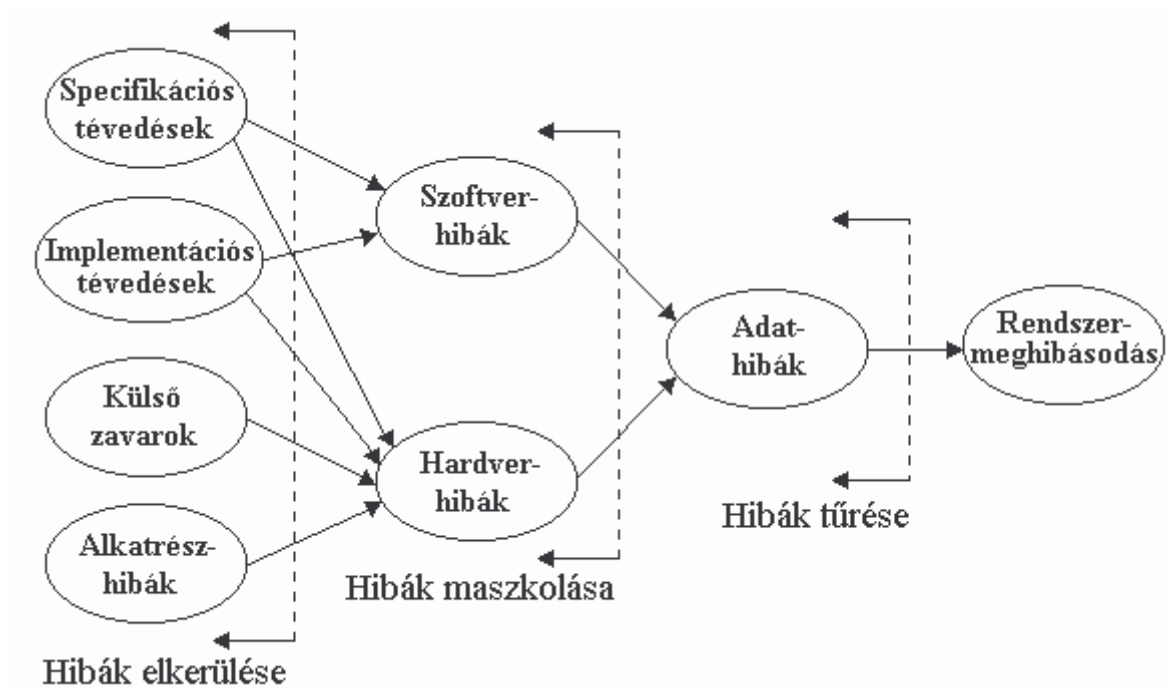
A működése közben a rendszer belső állapotok sorozatán megy át. Ha eközben fellép egy hibás átmenet, akkor a rendszer helytelen állapotba megy át, ami a további állapotváltások során a rendszer meghibásodásához vezethet. Egy rossz értéket a rendszer belső állapotában – például egy változó egy vagy több bitje fordított értéken áll – **adathibának** (*error*) nevezünk.

A helytelen állapotba való jutás oka lehet egy alkatrész meghibásodása (előre látható esemény), egy tervezési hiba a hardverben vagy a szoftverben, illetve egy kivitelezési hiba a megépítésben (előre nem látható események). A **hiba** (*fault*) tehát egy fizikai romlás a hardverben, vagy egy hiányosság a rendszer hardver vagy szoftver részében.

A rendszer valamely pontján fellépő hiba okozza az adathibát, ami a részegységek közötti adatcsere folytán elterjedhet, és végül a külső megfigyelő számára a rendszer meghibásodását eredményezheti:



A különböző hibák megnyilvánulása figyelembevételével határozhatjuk meg magas megbízhatóság elérésére alkalmazott eljárások korlátait (lásd az alábbi ábrát). A **hibatűrő mechanizmusok feladata** a hibákat még azelőtt felismerni, hogy a rendszer meghibásodásához vezetnének, és ennek megfelelően az adatokat helyes állapotba hozni, illetve – ha szükséges – kiiktatni a rendszerből a meghibásodott elemeket (újrakonfigurálás – *reconfiguration*).



A hibatűrő rendszerek néhány jellegzetes alkalmazási területe:

- **Kritikus rendszerek** az emberi életre, a környezetre, vagy értékes berendezésekre nézve, amelyek magas biztonságot (*safety*) igényelnek – űrhajók, repülőek vezérlőrendszerei, vegyi vagy nukleáris folyamatok vezérlőrendszerei, intenzív terápiás egészségügyi berendezések
- **Hosszú élettartammal rendelkező rendszerek** – személyzet nélküli űrhajók, műholdak
- **Késleltetett karbantartású rendszerek** – repülőgépek, távfeldolgozó rendszerek
- **Magas rendelkezésre állású rendszerek** – szerverek, elosztott tranzakciós (például banki) rendszerek
- **Nagy teljesítőképességű rendszerek** – szuperszámítógépek, párhuzamos számítógépek

8.2. A redundancia formái

Redundancia alatt a rendszerhez hozzáadott össze pótlólagos elemet értjük, amelyekre a funkciók ellátásához nem lenne szükség egy hibamentes világban. Még Neumann János kimutatta, hogy a hibatűrő képesség eléréséhez redundanciára van szükségünk. Ennek több formája van:

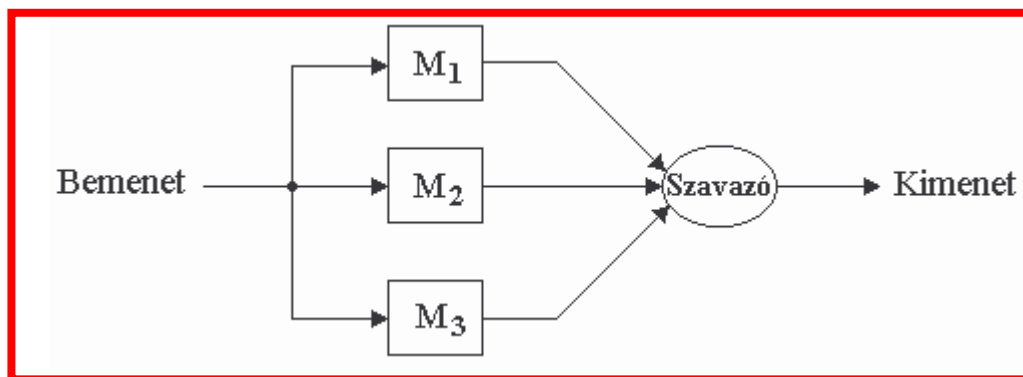
- **Fizikai redundancia** (*physical redundancy*) – a hardver vagy szoftver erőforrások sokszorozását (*replication*) feltételezi a hibák felismerése és tolerálása céljából. Az alapgondolat alternatívák párhuzamos feldolgozásából, majd az eredmények összehasonlításából, esetleg többségi szavazás alá vételéből áll. Nevezik még **strukturális redundanciának** is.
- **Információs redundancia** (*informational redundancy*) – a hasznos adatokhoz pótlólagos adatok hozzáadását feltételezi úgy, hogy kódolás által lehetővé váljon a hibák detektálása és esetleg kijavítása.
- **Időbeli redundancia** (*temporal redundancy*) – a feldolgozáshoz pótlólagos idő hozzáadását feltételezi, aminek a végrehajtás megismétlése céljából van szükség. Első sorban átmeneti hibák kezelésére hasznos.

8.3. Hardver-redundancián alapuló technikák

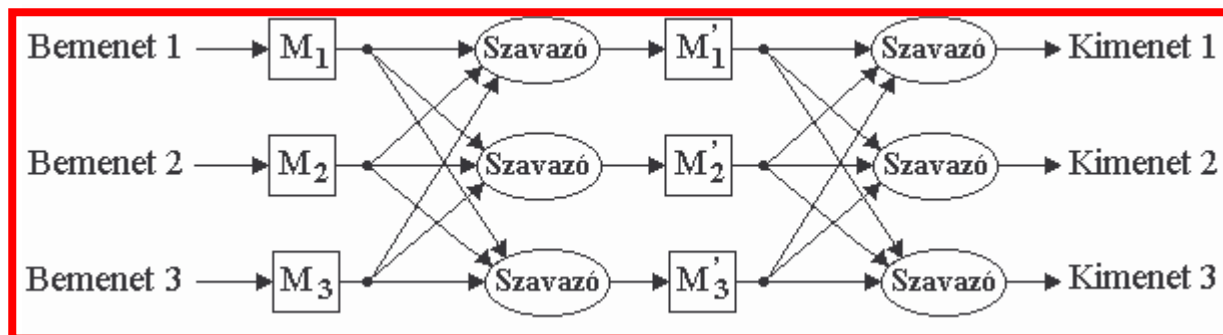
8.3.1. Statikus redundancia

A statikus redundancia alapelve a **hibák maszkolása** (*fault masking*), ami így megakadályozza ezek átváltozását a rendszerben elterjedhető adathibákká. A technika lényege az erőforrás sokszorozásából áll, ami egy többségi szavazás követ.

A legegyszerűbb konfiguráció a **hármass moduláris redundancia** (TMR – *Triple Modular Redundancy*), amely három azonos modulból áll (lásd az alábbi ábrát). A helyes működéshez elég legalább két működőképes modul, mert a kimeneti értékeket a szavazó összehasonlítja, és a többségnek megfelelő értéket választja ki. A rendszer egy modul meghibásodását tolerálja.



Hogy a szavazó meghibásodása ne vezessen a rendszer összeomlásához, kidolgozták a szavazó áramkör megháromszorozásával megépített rendszert (ez az adatútvonalak megháromszorozását is feltételezi):

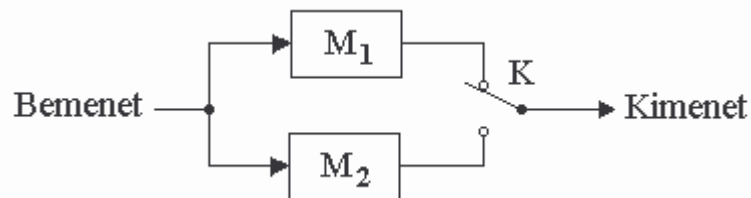


A TMR általánosítása az **N-moduláris redundancia** (NMR – *N-Modular Redundancy*), amely az alapmodul N darab másolatát használja (N páratlan szám). Ha $N=2p+1$, akkor a rendszer p modul vagy szavazó meghibásodását képes tolerálni.

A statikus redundanciát első sorban a kritikus rendszerekben alkalmazzák, ahol egy helytelen eredmény szolgáltatása – akár csak kis ideig – elfogadhatatlan. A hiba maszkolásához viszont jelentős plusz hardverre van szükség, ami ezt a technikát költségessé teszi.

8.3.2. Dinamikus redundancia

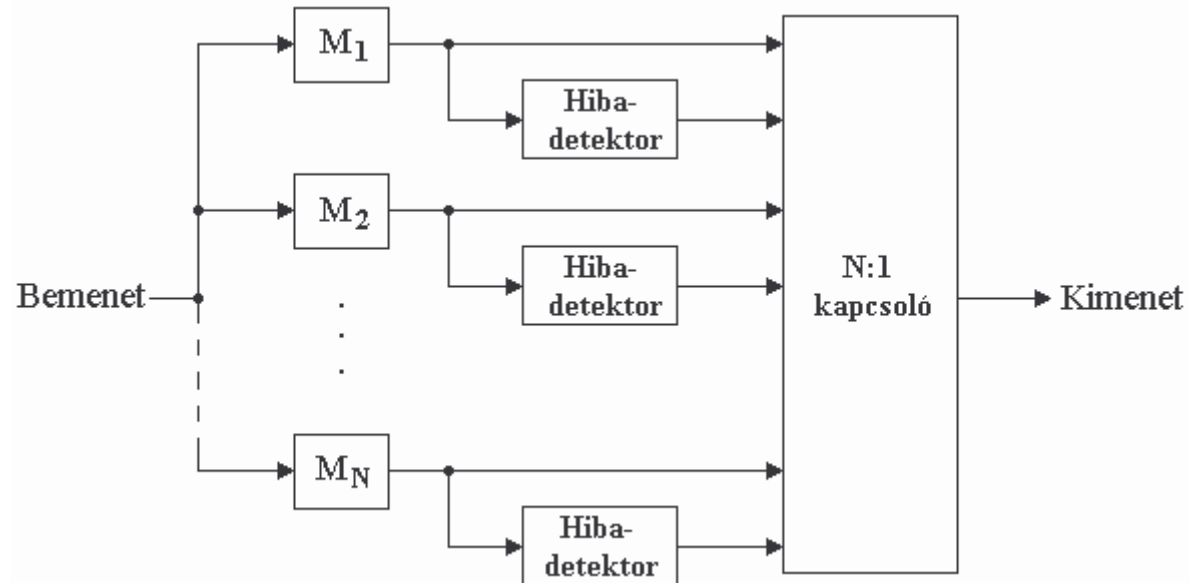
Ez a típusú redundancia, egy hiba felismerése után, a rendszer kimenetének egy tartalék (*standby*) modulra való automatikus átkapcsolására épül (a kapcsolót egy digitális áramkör valósítja meg):



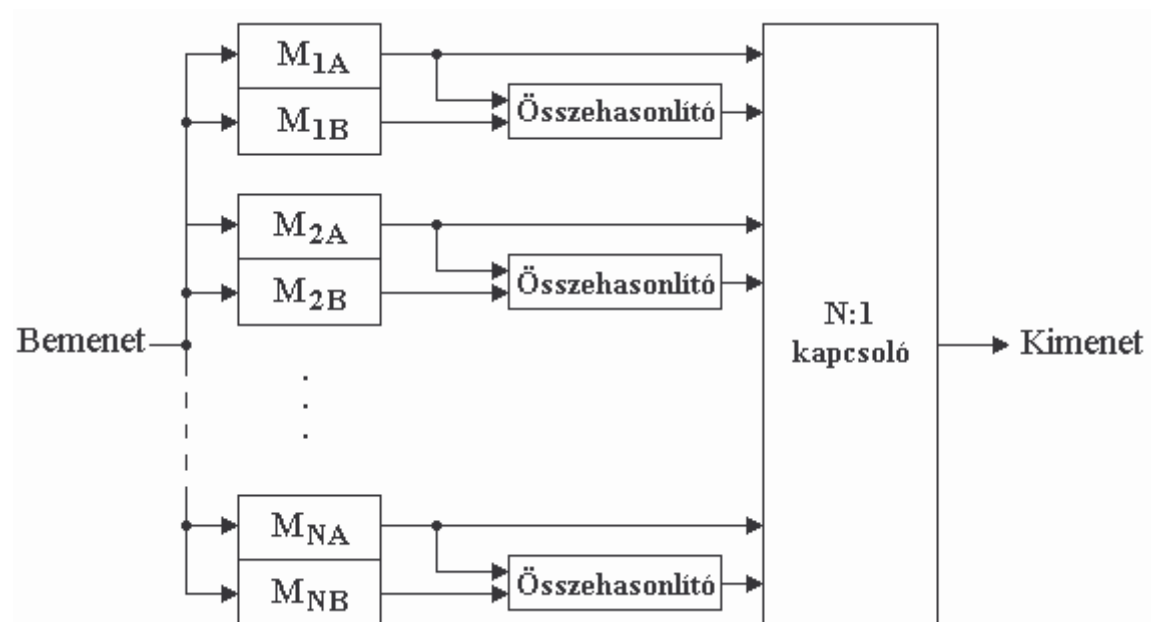
A megfelelő működés végett, szükséges egy hatékony hibadetektálási módszer implementálása. Erre több lehetőség van, mint a kimeneti érték abszolút tesztelése (például az eredmény határértékek között való illeszkedése), hibadetektáló kód alkalmazása, vagy *watchdog* (periodikusan újraindított időzítő) használata.

Íme két példa dinamikus redundanciát használó rendszerre:

- A **standby sparing** rendszerben az N egyforma modul egyike szolgáltatja a kimenetet, a többi $N-1$ készenléti tartalék. Ez a rendszer $N-1$ modul meghibásodását képes tolerálni. A megfelelő működést nagymértékben meghatározza a hibadetektor hatékonysága.



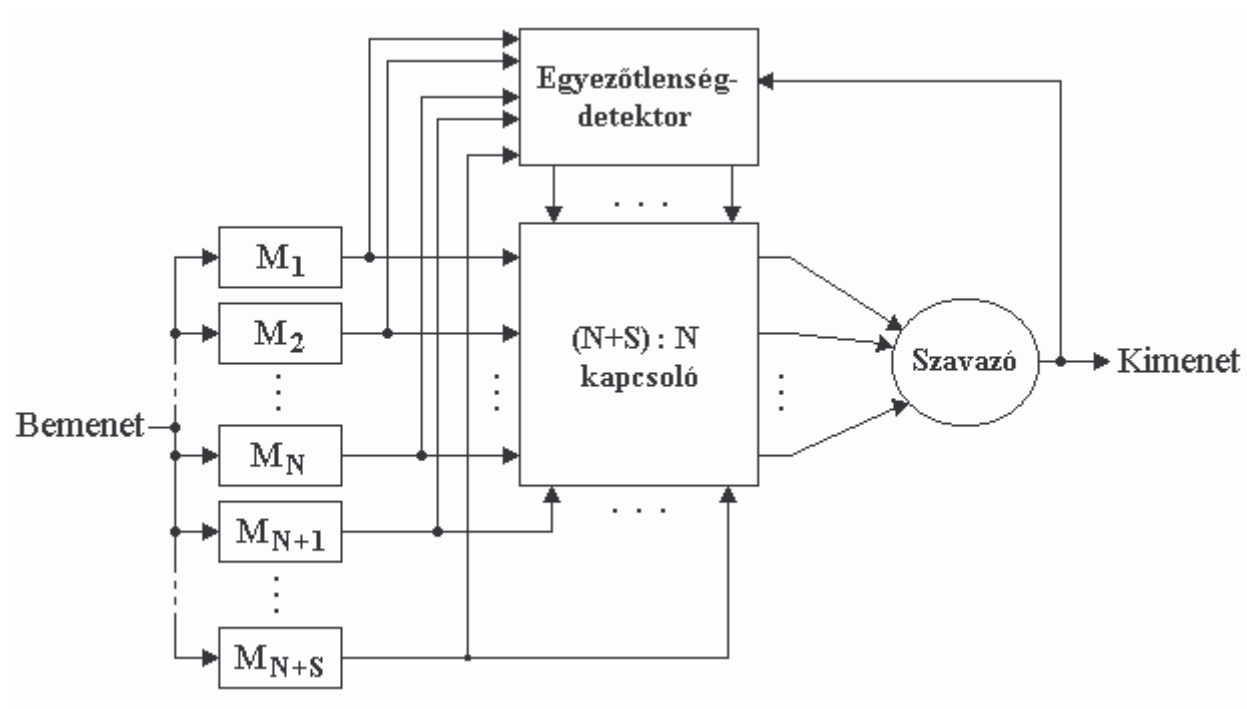
- A **pair-and-spare** rendszerben egyidejűleg két aktív modul van, amelyeknek az eredményeit összehasonlítják. Ha eltérés van, akkor az egész modulpárt kicseréli a kimeneti kapcsolóáramkör egy másikkal, a tartalékok közül. Ez a megoldás költségesebb, mint az előző, de az összehasonlítás alapú hibadetektálásnak igen magas a hatékonysága tetszőleges hibák esetén is.



Általában a dinamikus redundanciát a hosszú élettartammal rendelkező, valamint a magas rendelkezésre állást igénylő rendszereknél alkalmazzák, ahol ideiglenesen el lehet fogadni a helytelen kimenetet, de a működőképes állapotot hamar vissza lehet állítani a újrakonfigurálás által. Ez a technika kevesebb pótlólagos hardvert használ, mint a statikus, ezért annál olcsóbb.

8.3.3. Hibrid redundancia

A hibrid redundancia alapelve a **tartalékos N-moduláris redundanciában** (*NMR with spares*) testesül meg (lásd az alábbi ábrát). Egy N egyforma modulból álló mag mindig részt vesz a szavazásban, és meghatározza a rendszer kimenetét. Egy modul meghibásodása esetén megtörténik a rendszer újrakonfigurálása: a kapcsoló hálózat kicseréli a meghibásodott modult egy tartalékkal. Ha a rendszer $N+S$ modulból áll ($N=2p+1$), akkor maximálisan $p+S$ modul hibáját tudja tolerálni.



KÉRDÉSEK, TÉTELEK

- 8.1. Mik a számítógépek megbízhatósága növelésének módszerei, és ezeket mikor kell alkalmazni?
- 8.2. Melyek a hibatűrő rendszerekben alkalmazott redundancia formái, és ezek legfontosabb jellemzői?
- 8.3. Vázolja fel a TMR (hármass moduláris redundancia) rendszert, és magyarázza el a működését!
- 8.4. Vázolja fel a *standby sparing* architektúrát, és magyarázza el a működését!
- 8.5. Vázolja fel a *pair-and-spare* architektúrát, és magyarázza el a működését!

9. MODELLEZÉS ÉS SZIMULÁCIÓ.

[Dr. Szász Gábor, Dr. Kun István és Dr. Zsigmond Gyula jegyzetei alapján]

9.1. A MARKOV-folyamatok elmélete

A hibatűrő rendszerek alkalmazásának elsődleges célja a megbízhatóság növelése azáltal, hogy akár meghibásodott alkotóelem esetében is a rendszer működőképes maradhat. Az elektronikai (informatikai) rendszerek megbízhatósági vizsgálatainál széleskörű alkalmazási lehetőséget nyújt a véges állapotú, folytonos idejű MARKOV-folyamatok elméletére támaszkodó modellezés; megkötést tulajdonképpen csak az exponenciális eloszlás feltételezése jelent. Ez a feltételezés azonban a komplex villamos rendszerekben használt eszközök jelentős részénél nem jelent különösebb megszorítást.

Egy $X(t)$ sztochasztikus folyamat folytonos idejű MARKOV-lánc, ha minden egész n szám esetén tetszőleges olyan $t_1 \dots t_{n+1}$ sorozatra, amelyre $t_1 < \dots < t_{n+1}$, teljesül, és eleget tesz az alábbi összefüggésnek:

$$P[X(t_{n+1}) = j | X(t_1) = i_1, X(t_2) = i_2, \dots, X(t_n) = i_n] = P[X(t_{n+1}) = j | X(t_n) = i_n]. \quad (9.1)$$

A (9.1) kifejezése tükrözi a MARKOV-folyamatoknak azt az alapvető tulajdonságát, hogy a múlt befolyását a folyamat jövőjére a pillanatnyi állapot tartalmazza. A folyamat „emlékezet nélküli”. A folytonos idejű MARKOV-lánc esetében bármely időpillanatban bekövetkezhet állapotváltozás. A rendszer vizsgálatához szükségünk van annak megítéléséhez, hogy a folyamat mennyi ideig van egyik állapotában, mielőtt átlépne egy másik állapotba. Esetünkben ennek az időtartamnak exponenciális eloszlásúnak kell lennie (*markov-tulajdonság*).

A MARKOV-láncok állapotait a

$$\frac{dP_i(t)}{dt} = a_{ii}(t) \cdot P_i(t) + \sum_j a_{ij}(t) \cdot P_j(t) \quad (9.2)$$

közönséges differenciál-egyenlet írja le, ahol $P_i(t)$ annak a valószínűsége, hogy a folyamat a t időpontban az i állapotban van, hasonlóan értelmezendő a P_j is.

Az $a_{ij}(t)$ intenzitást a következő összefüggés definiálja:

$$a_{ij} = \frac{p_{ij}(t, t + dt)}{dt}. \quad (9.3)$$

A $p_{ij}(t_n, t_{n+1})$ átlépési valószínűséget a

$$p_{ij}(t_n, t_{n+1}) = P[X(t_{n+1}) = j | X(t_n) = i], \quad (9.4)$$

képlet adja meg. Az $a_{ii}(t)$ kifejezhető $a_{ij}(t)$ segítségével:

$$a_{ii}(t) = - \sum_j a_{ij}(t). \quad (9.5)$$

Ha a $p_{ij}(t_n, t_{n+1})$ nem függ a t_n időponttól, a MARKOV-lánc *homogén*. Amennyiben egy MARKOV-lánc homogén, véges számú állapota van és az átlépési valószínűségek pozitívak, a MARKOV-lánc *ergodikus*.

9.2 MARKOV-féle megbízhatósági modell

Az informatikai rendszerek megbízhatósági vizsgálatainál széleskörű alkalmazási lehetőséget nyújt a véges állapotú, folytonos idejű MARKOV-folyamatok elméletére támaszkodó modellezés. Megkötést tulajdonképpen csak az exponenciális eloszlás feltétele jelent, ez azonban az informatikai rendszerek jelentős részénél teljesül. A modell jellemzői:

- A rendszer egy sztochasztikus folyamat szerint különböző állapotokba kerülhet.
- A modellezés során az egyes állapotok közötti átmenetek lehetőségéből indulunk ki.
- A rendszer állapotait a $t=0$ időponttól vizsgáljuk.

A modell nagy előnye, hogy lehetővé teszi javítható rendszerek számítását is. Természetesen a MARKOV-féle megbízhatósági modell alkalmazásánál is figyelembe kell venni a vizsgált technikai rendszer speciális tulajdonságait is. Mindenképpen célszerű megvizsgálni, hogy a technikai rendszernek nincs-e valamilyen tanulási algoritmus, önszerveződő képessége.

A MARKOV-folyamatokkal modellezett technikai rendszerek megbízhatósági vizsgálatainál elterjedt a gráfelméleti módszerek alkalmazása. A jelfolyam-gráfokat különösen eredményesen lehet használni az MTTF kiszámításához. Ezért a következő pontban összefoglaljuk a gyakorlati alkalmazásokhoz szükséges jelfolyam-gráfokkal kapcsolatos ismereteket.

9.2.1 Jelfolyam-gráfok

A jelfolyam-gráf olyan eszköz, amellyel ábrázolhatók egy lineáris algebrai egyenletrendszer változói között fennálló ok és okozati kapcsolatok. A kapcsolatok feltárása nagymértékben elősegítheti egy adott rendszer tulajdonságainak megismerését és esetenként a szükséges matematikai műveletek elvégzését.

A jelfolyam-gráfok élekből és csúcsokból állnak. A csúcs itt bármilyen változót jelenthet, de összegző és átvivő tulajdonsága is van. Az élhez átviteli tényezőket, vagy függvényeket rendelünk. A jelfolyam-gráfokkal kapcsolatos további megnevezések és definíciók:

- *Bemenő csúcsoknak* (forrásnak) csak kifutó ágai vannak (pl. a 9.1 ábra x_1 csúcsa).
- *Kimenő csúcs* (nyelő) olyan csúcs, amelynek csak befutó ágai vannak (pl. az x_5 csúcs). Ha olyan csúcsot kell kimenő csúcsnak tekinteni, amelyre nem teljesül a fenti feltétel egy egységnyi átviteli tényezőjű éllel csatlakozunk a kérdéses csúcshoz, és pótlólagos változót (kimenő csúcsot) vezetünk be.
- Az *út* egymáshoz folyamatosan csatlakozó azonos irányban bejárt élek bármely sorozata.
- Az *előrevezető út*, olyan út, amely egy bemenő csúcstól egy kimenő csúcsig vezet úgy, hogy minden érintett csúcson csak egyszer halad át.
- A *hurok* olyan út, amely ugyanabból a csúcsból indul, ahová befut, és az érintett csúcsokon csak egyszer halad át.
- Az *út átviteli tényezője* az út befutása során érintett élek átviteli tényezőinek szorzata.

- Az *előrevezető út átviteli tényezője* az előrevezető út befutása során érintett élek átviteli tényezőinek szorzata.

A *huroktényező* a hurkot alkotó út átviteli tényezője.

Egy adott egyenletrendszer, amely $j=1,2,\dots,N$ egyenletből áll ok és okozati formában az

$$X_j = \sum_{k=1}^N a_{kj} x_k \quad (9.6)$$

alakban írható fel. Abban az esetben, ha egy lineáris rendszert integro-differenciálegyenletek írnak le, a jelfolyam-gráf alkalmazásához először LAPLACE-transzformációt kell végrehajtani. A gyakorlati vizsgálatoknál a (9.6) összefüggést sokszor célszerű az alábbiak szerint értelmezni.

$$\textbf{kimenőjel} = (\textbf{átviteli tényező}) \cdot (\textbf{bemenőjel})$$

Egy adott egyenletrendszer jelfolyam-gráfjának megszerkesztése lényegében az ok és okozati összefüggések végigkövetését jelenti: mindegyik változót önmagával és más változókkal kifejezve kapcsolatban hozunk.

Példaképpen rajzoljuk fel az alábbi egyenletrendszer jelfolyam-gráfját, független

változók: x_1 és x_2 .

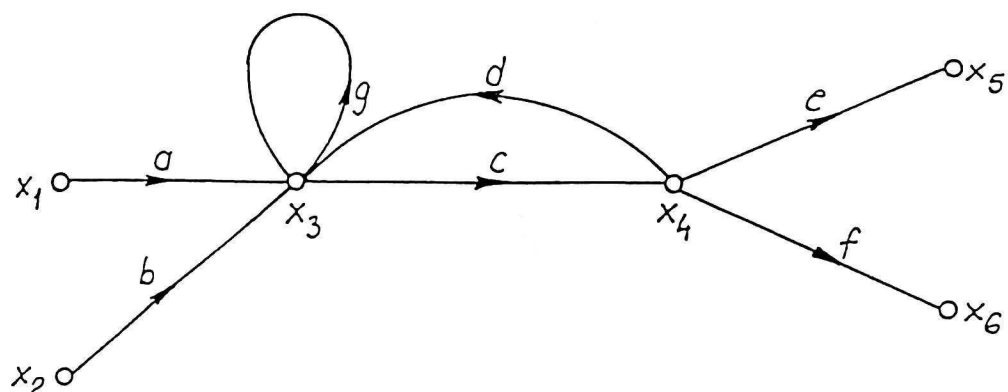
$$x_3 = ax_1 + bx_2 + dx_4 + gx_3$$

$$x_4 = cx_3$$

$$x_5 = ex_4$$

$$x_6 = fx_4$$

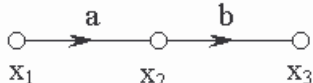
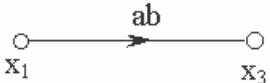
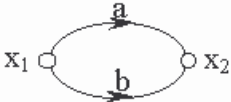
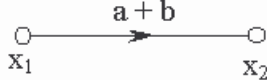
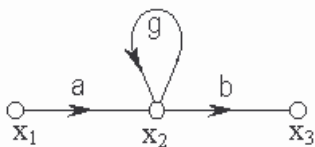
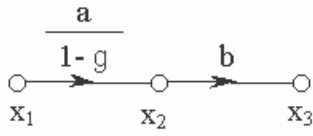
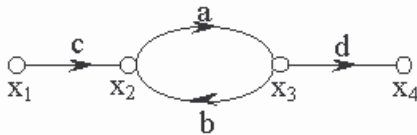
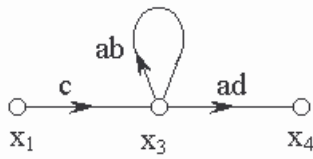
A jelfolyam-gráf a 9.1 ábrán látható.



9.1. ábra. Egy jelfolyam-gráf

Az ok és okozati kapcsolatok tisztázása után rendszerint felmerül a kérdés, milyen összefüggés áll fenn két kiválasztott lényeges változó (jel) között. Ezt a feladatot a gyakorlatban legtöbbször a jelfolyam-gráf egyszerűsítésével oldhatjuk meg. Bonyolult esetekben számítógépes módszereket kell alkalmazni.

A jelfolyam-gráfok egyszerűsítésének legfontosabb szabályai a 9.2 ábrában találhatók meg.

Megnevezés	Eredeti alakzat	Átalakított alakzat
Közbenső csúcs kiküszöbölése		
Párhuzamos élek összevonása		
Saját hurok kiküszöbölése		
Csúcs kiküszöbölése		

9.2. ábra. Egyszerűsítési szabályok

9.2.2 Jelfolyam-gráfok alkalmazása

A jelfolyam-gráfokat leggyakrabban az MTTF kiszámítására alkalmazzuk. Az alábbiakban e a fontos mutató számításának menetét mutatjuk be. Először felrajzoljuk a jelfolyam-gráfot. A jelfolyam-gráf felrajzolása az alábbiak szerint történik:

- Az N különböző állapotnak N különböző csúcs felel meg.
- A csúcsokat összekötő éleket az állapotokból történő átlépéseknek megfelelően a szükséges a_{ij} intenzitás ($i \neq j$) LAPLACE-transzformáltjával súlyozzuk. Ha $i > j$, az intenzitás λ . Ha $j > i$, az intenzitás μ .

- c. A csúcsoknál hurkot képezünk, melyet az elmenő élek intenzitás összegének LAPLACE-transzformáltjával súlyozunk és azt (-1) -el szorozzuk.
- d. A kezdeti feltételeket a kiindulási állapothoz tartozó $\frac{1}{s}$ súlyozású éllel kell figyelembe venni.

A felrajzolt jelfolyam-gráfot ezután, átalakítjuk úgy, hogy a hurkokat és a csúcsokat eltüntetjük, a hibás állapothoz tartozó k csúcs kivételével, az így kapott $Z_k(s)$ függvény a hibás állapotban való tartózkodás valószínűségének LAPLACE-transzformáltja. A $Z_k(s)$ kiszámításához a reális üzemi feltételek számára tervezett rendszerek esetében általában elégséges az egyszerűsítő módszerek alkalmazása.

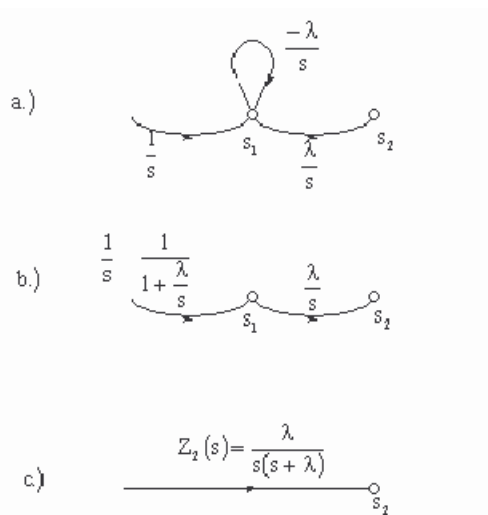
A $Z_k(s)$ ismeretében kiszámíthatjuk a megbízhatósági függvény $R(s)$ LAPLACE-transzformáltját:

$$R(s) = \frac{1}{s} - Z_k(s) \quad (9.7)$$

Kihasználva az improprius integrálok és a LAPLACE-transzformáció aszimptotikus tulajdonságait, felírható, hogy

$$MTTF = \lim_{s \rightarrow 0} \left(\frac{1}{s} - Z_k(s) \right) \quad (9.8)$$

Illusztratív példaként jelfolyam-gráf segítségével számítsuk ki egy nem javítható rendszer $MTTF$ értékét! A meghibásodási ráta: λ . Először a 9.3 ábrán látható módon meghatározzuk a $Z_2(s)$ függvényt.



9.3. ábra. Nem javítható elem jelfolyam-gráfja

(9.7) szerint a megbízhatósági függvény LAPLACE-transzformáltja:

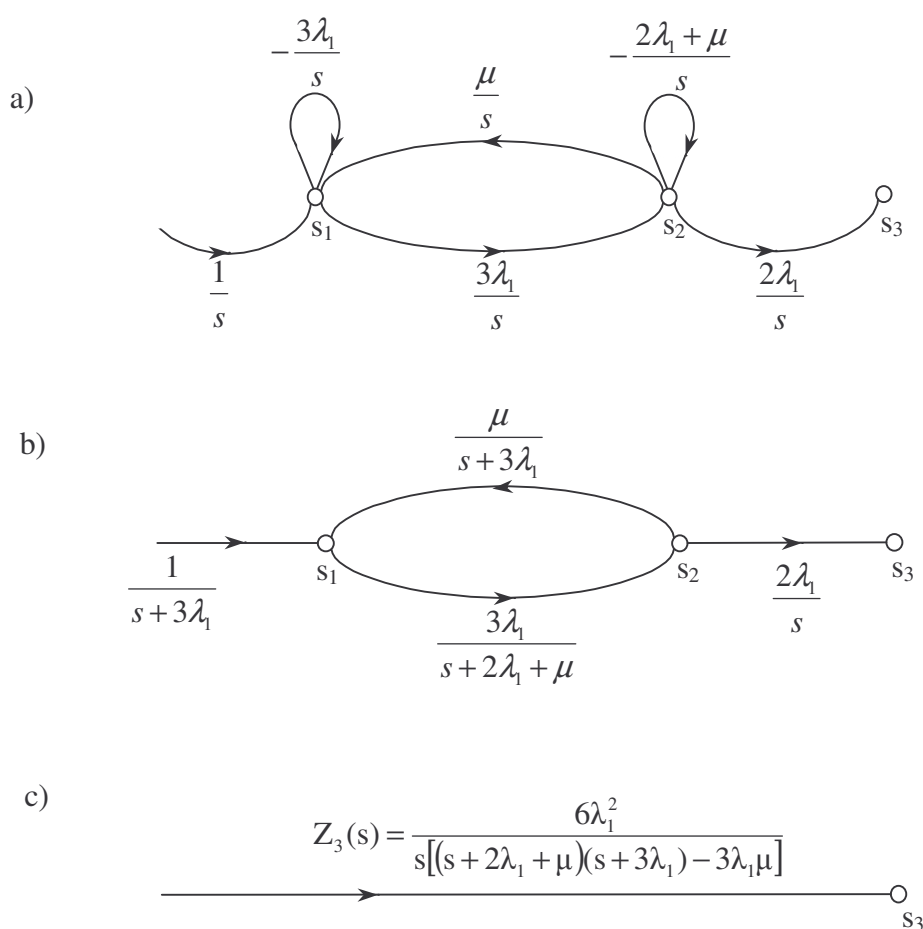
$$R(s) = \frac{1}{s} - Z_2(s) = \frac{1}{s} - \frac{\lambda}{s(s + \lambda)} = \frac{1}{s + \lambda}$$

Így megkapjuk az előzőekből jól ismert eredményt: $MTTF = \lim_{s \rightarrow 0} \frac{1}{s + \lambda} = \frac{1}{\lambda}$.

9.2.3 MARKOV-féle modell alkalmazása

A MARKOV-féle modell alkalmazásának illusztrálására számítsuk ki egy olyan rendszerre az $MTTF$ értékét, amely három alrendszerből áll. A rendszert akkor tekintjük jónak, ha legalább két alrendszer üzemel. Amennyiben valamelyik alrendszer meghibásodik, akkor kezdődik el a javítás. A javítás befejezésével a rendszer visszatér előző állapotába, hacsak a javítás ideje alatt meg nem hibásodik a tartalék. A részrendszerek meghibásodási rátája λ_1 , javítási rátája μ .

Először felrajzoljuk a jelfolyam-gráfot (9.4. a. ábra) majd elvégezzük a szükséges átalakításokat (9.4. b. és c. ábrák).



9.4. ábra. Az $MTTF$ számításához szükséges jelfolyam-gráfok

A $Z_3(s)$ ismeretében:

$$R(s) = \frac{1}{s} - Z_3(s) = \frac{s + 5\lambda_1 + \mu}{s^2 + s(5\lambda_1 + \mu) + 6\lambda_1^2}.$$

Így (9.8) alapján:

$$MTTF = \lim_{s \rightarrow 0} R(s) = \frac{5\lambda_1 + \mu}{6\lambda_1^2}.$$

9.3. Az állapotfüggő karbantartás kibernetikai alapjai

Tegyük fel, hogy van valamilyen műszaki berendezésünk, amelynek viselkedését az alábbi *rendszeregyenlet* írja le:

$$\Phi(\mathbf{x}, \mathbf{u}, t) = \mathbf{0} \quad (9.9)$$

a kimeneti folyamatot pedig az alábbi kimeneti egyenlet:

$$\mathbf{v} = \Psi(\mathbf{x}, \mathbf{u}, t) \quad (9.10)$$

($\mathbf{x} \in \mathfrak{X}_n$, $\mathbf{u} \in \mathfrak{X}_m$ és $\mathbf{v} \in \mathfrak{X}_k$. $t \in \mathfrak{T}$ a rendszeridő. Φ és Ψ valós, vektorértékű operátorok az $n+m+1$ dimenziós időfüggvények terén.)

$\mathbf{u}$ a rendszer bemenőjele, amely az irányító és a zavaró jeleket is tartalmazza.

$\mathbf{v}$ a kimenőjel, vagyis a rendszer válasza.

$\mathbf{v}$ kiértékelése után új beavatkozó jel képződik, amely megpróbálja a rendszert „jobb” kimenőjel adására készíteni. A bemenőjel zajos is lehet, a rendszer sztochasztikusan is működhet.

Legyen a rendszer állapotvektora felbontható az alábbi formában:

$\mathbf{x} = (\mathbf{y}^T, \mathbf{w}^T)^T$, ahol $\mathbf{y} \in \mathfrak{X}_q$ jellemzi a műszaki állapotot vagy *kondíciót*,

$\mathbf{w} \in \mathfrak{X}_{n-q}$, pedig a működési állapotot vagy *státuszt*.

Legyen adva $\mathfrak{X}_q$ diszjunkt részhalmazainak egy osztálya, amelyeket $Y_1, Y_2, \dots, Y_r$ jelöl.

Ezen részhalmazok mindegyike megfelel a rendszer valamilyen *minőségi osztályának* (előírási állapot, degradált állapot stb.). Mindegyik minőségi osztálynak lehet saját rendszeregyenlete:

$$\Phi_i(\mathbf{x}, \mathbf{u}) = \mathbf{0} \quad i = 1, 2, \dots, r \quad (9.11)$$

és kimeneti egyenlete:

$$\mathbf{v} = \Psi_i(\mathbf{x}, \mathbf{u}) \quad i = 1, 2, \dots, r \quad (9.12)$$

ahol Φ_i és Ψ_i valós, vektorértékű operátorok a valós $n+m$ dimenziós időfüggvények terén. (9.11) és (9.12) abban különbözik (9.10) és (9.11)-től, hogy eltűnt az explicit időfüggés.

A minőségi osztályok közti **átmenetnek** két formája lehetséges:

kényszerátmenet

(pl. javítás)

véletlen átmenet

(pl. meghibásodás)

Legyen $\mathbf{x}(t) = [\mathbf{y}(t)^T, \mathbf{w}(t)^T]^T$ a rendszerállapotot leíró időfüggvény, és definiáljuk az alábbi eseményt:

$$A_i(t) = \{\mathbf{y}_i(t) \in Y_i\} \quad (9.13)$$

Bevezetjük a λ_i $i=1, 2, \dots, r$ és y_{ij} , $i, j=1, 2, \dots, r$ nem negatív számokat a következő átmenetmodellhez:

$$P[A_i(t + \Delta t) | A_i(t)] = 1 - \lambda_i \Delta t + o(\Delta t) \quad (9.14)$$

$$P[A_j(t + \Delta t) | A_i(t)] = \lambda_i y_{ij} \Delta t + o(\Delta t) \quad (9.15)$$

Ha egy minőségi osztályt a rendszer képtelen elhagyni, akkor a λ_i állapot-elhagyási ráta értéke nullával egyenlő.

Megjegyzés: a nagy, közepes és kicsiny megbízhatóság megfelel a kicsiny, közepes és nagy $\lambda_i > 0$ értékeknek.

A rendszert *osztályozhatónak* nevezzük, ha $\mathbf{v}$ trajektóriájából meghatározható az aktuális minőségi osztály valamilyen osztályozási algoritmus segítségével egy olyan valószínűségi szinten, amely valamilyen értelemben kielégítőnek tekinthető. Legyen az osztályozás eredménye a j osztályindex. Ha elégedettek vagyunk, továbblépünk, de ha nem, akkor kényszerítjük a rendszert, hogy osztályt változtasson (pl. javítás útján). Az **állapotfüggő karbantartás** éppen ezt a funkciót látja el.

Ezen a ponton elvégezhetjük a rendszer „finomhangolását” $\mathbf{u}$ értékének megfelelő módosításával. Ezután ismét megfigyeljük $\mathbf{v}$ -t.

Minden osztályváltási, és finomhangolási tevékenységnek, a matematikai alakfelismeréssel együtt, van valamekkora **költség**, míg a kimeneti vektor kívánatos értékei **fedezetet** termelnek, a nemkívánatosak pedig ismét csak költséget (vagy elmaradó hasznot).

(Költségfüggvény és célfüggvény-konstrukciót illetően lásd a szakirodalmat.)

Kibernetikai értelemben egy szabályozott **MARKOV-folyamatot** kell optimalizálnunk, vagyis **markovi döntésfolyamatot** kapunk.

Általában nem közvetlenül a kimeneti vektor, hanem több megfigyelésből származó adat alapján osztályozunk. Ezért kell **dimenziócsökkentést** alkalmazni. Ez a **lényegkiemelés** statisztikai, vagy műszaki jellegű.

9.4. Kidolgozott példák

1. Folyamatos gyártóvonalon készülő terméket automatikusan ellenőriznek. Az ellenőrző műszer meghibásodása esetén azonnal átkapcsolnak egy tartalék műszerre, amely állandó készenlétben van (hideg tartalék). A rendszer megbízhatóságának jobb modellezése érdekében feltételezünk még az átkapcsolásnál bizonyos mértékű hibalehetőséget, és azt, hogy az első műszert meghibásodásakor – a hiba helyének felderítése után – azonnal javítani kezdik, így az egy idő múlva ismét üzembe helyezhető. A rendszer eredő megbízhatóságának megállapításához az alábbi állapotok definiálhatók:

S_1 : Az egyik műszer **működik**, a másik tartalékban van.

S_2 : Az egyik műszer meghibásodott, átkapcsoltunk a tartalék egységre.

Megkezdődik a hiba keresése. A rendszer **működik**.

S_3 : A hiba lokalizálása után megkezdődött a javítás. A tartalék egység végzi a mérést. A rendszer **működik**.

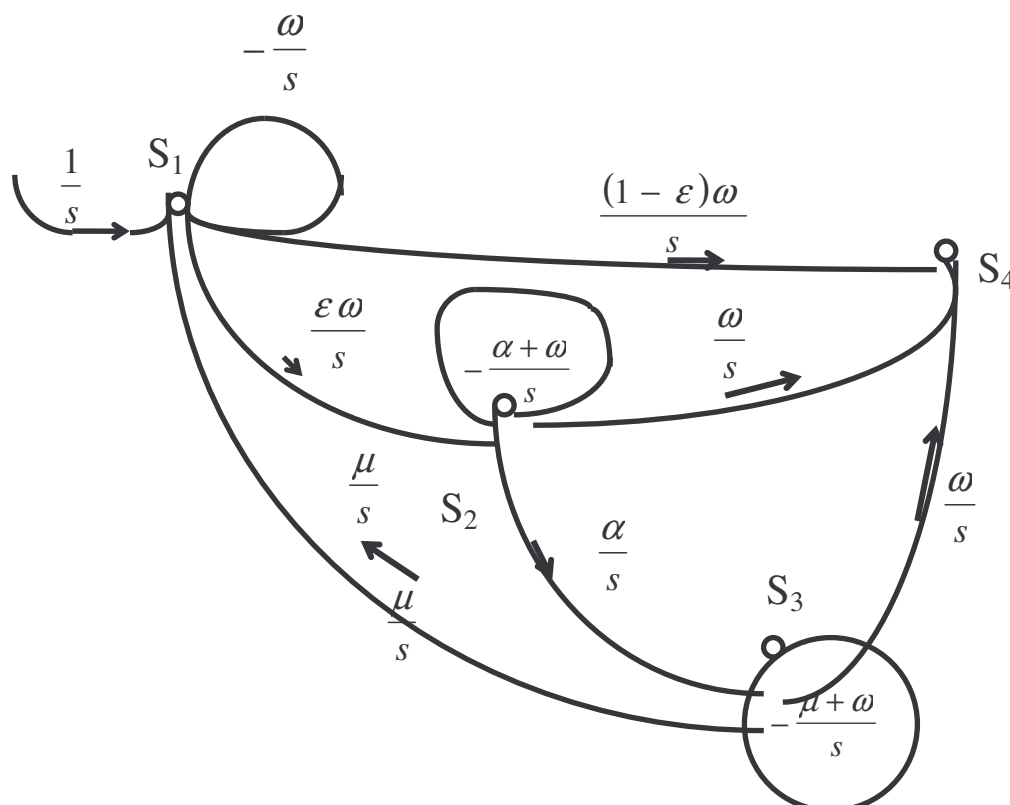
S_4 : Hibás átkapcsolás következtében nem indult meg, illetve a tartalék egység meghibásodása miatt megszűnt a mérés, még mielőtt az eredetileg üzemelő műszert helyreállították volna. A rendszer **nem működik**.

Legyen minden műszerre ω a két meghibásodás közt eltelt átlagos idő reciproka (azonos típusú műszerek), ε az átkapcsolási művelet sikerességének valószínűsége, α a hibakeresés várható időtartamának reciproka és μ a javításhoz átlagosan szükséges idő reciproka. Legyen pl.: $\omega=10^{-3}[\text{h}^{-1}]$, $\varepsilon=0,995$, $\alpha=1[\text{h}^{-1}]$ és $\mu=0,5[\text{h}^{-1}]$!

- Készítsük el a megbízhatósági modell jelfolyam-gráfját!
- Határozzuk meg a rendszer első leállításának várható idejét!
- Viszonyítsuk ezt egyetlen műszer *MTTF*-jéhez!

A válasz részletes kidolgozása

A megbízhatósági modell jelfolyam-gráfja:



Az $\frac{1}{s}$ bemenő él azt jelzi, hogy a folyamat az S_1 állapotból indul.

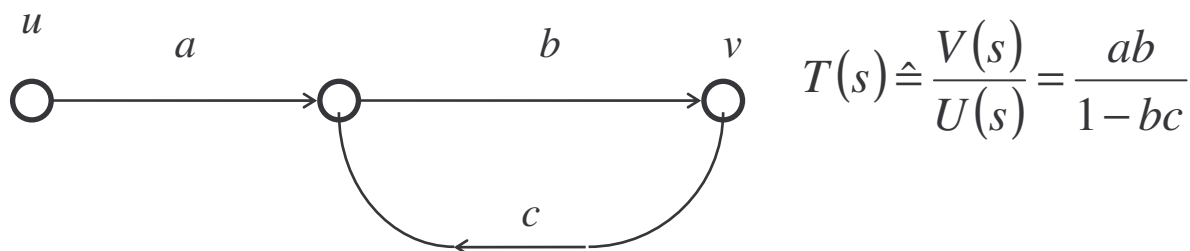
b. Mielőtt a jelfolyam-gráf alapján meghatároznánk a működő állapotokban tartózkodás valószínűségének a LAPLACE-transzformáltját, érdemes egy kis elméleti kitérőt tenni. (9.8) alapján:

$$M(\xi) = \lim_{s \rightarrow 0} s \mathcal{L} \left\{ \int_0^t R(x) dx \right\} = \lim_{s \rightarrow 0} R(s)$$

Tehát a várható élettartam a megbízhatósági függvény LAPLACE-transzformáltjának $s \rightarrow 0$ határértékével egyenlő. Ezt a fontos eredményt fogjuk fölhasználni a továbbiakban.

A működő állapotokat összekötő éleket $\frac{\mu}{s}$ -nél fölbontjuk. Ekkor kiszámítjuk a hurokátviteli függvényt, melynél figyelembe kell venni a működő állapotoknál lévő

hurkokat is. A jelfolyam-gráfok algebrája szerint az alábbi összefüggést alkalmazhatjuk:



Tehát a hurokátviteli függvény:

$$Y(s) = \frac{1}{1 + \frac{\omega}{s}} \cdot \frac{\varepsilon \omega}{s} \cdot \frac{\frac{\alpha}{s}}{1 + \frac{\alpha + \omega}{s}} \cdot \frac{\frac{\mu}{s}}{1 + \frac{\mu + \omega}{s}} = \frac{\alpha}{(s + \omega)(s + \alpha + \omega)(s + \mu + \omega)}$$

Jelöljük az egyes állapotokban tartózkodás valószínűségének LAPLACE-transzformáltját rendre $X_i(s)$ -sel. Tudjuk, hogy

$$R(s) = \sum_{i=1}^3 X_i(s)$$

Most pedig meghatározzuk a jelfolyam-gráfok algebrájának ismételt alkalmazásával a működő állapotokban tartózkodás valószínűségeinek LAPLACE-transzformáltját:

$$X_1(s) = \frac{\frac{1}{1 + \frac{\omega}{s}} \cdot \frac{1}{s}}{1 - Y(s)} = \frac{1}{s + \omega};$$

$$X_2(s) = \frac{\frac{\varepsilon \omega}{s} \cdot \frac{1}{1 + \frac{\alpha + \omega}{s}}}{s} \cdot X_1(s) \text{ és}$$

$$X_3(s) = \frac{\frac{\alpha}{1 + \frac{\mu + \omega}{s}}}{s} \cdot X_2(s) = \frac{\alpha}{s + \mu + \omega} \cdot X_2(s).$$

Most már csak az összegzés és a határértékképzés van hátra:

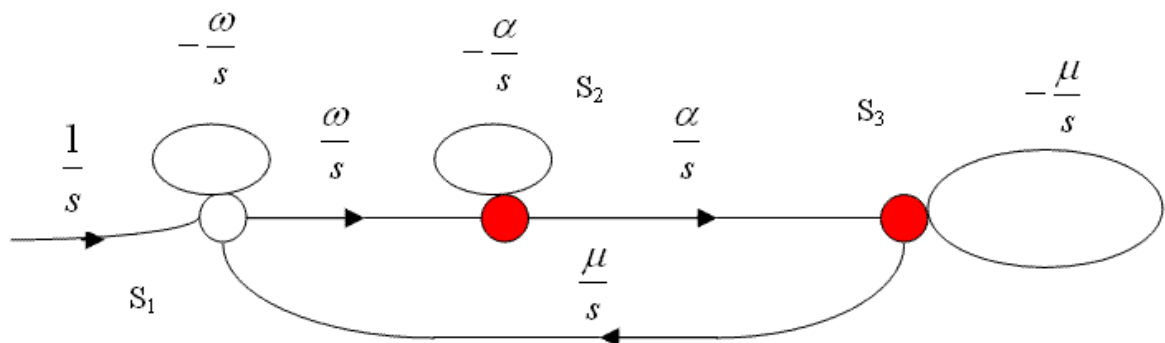
$$\begin{aligned}
 R(s) &= \frac{\frac{1}{s+\omega} \left[1 + \frac{\varepsilon\omega}{s+\alpha+\omega} \left(1 + \frac{\alpha}{s+\mu+\omega} \right) \right]}{1-Y(s)} = \\
 &= \frac{(s+\alpha+\omega)(s+\mu+\omega) \left[1 + \frac{\varepsilon\omega}{s+\alpha+\omega} \left(1 + \frac{\alpha}{s+\mu+\omega} \right) \right]}{(s+\omega)(s+\alpha+\omega)(s+\mu+\omega) - \alpha\varepsilon\mu\omega} = \\
 &= \frac{[s+\alpha+\omega(1+\varepsilon)](s+\mu+\omega) + \alpha\varepsilon\omega}{(s+\omega)(s+\alpha+\omega)(s+\mu+\omega) - \alpha\varepsilon\mu\omega}
 \end{aligned}$$

$$\begin{aligned}
 M(\xi) &= \lim_{s \rightarrow 0} R(s) = \frac{[\alpha + \omega(1+\varepsilon)](\mu + \omega) + \alpha\varepsilon\omega}{\omega[(\alpha + \omega)(\mu + \omega) - \alpha\varepsilon\omega]} = \frac{1}{\omega} \cdot \frac{1,001995 \cdot 0,501 + 0,000995}{1,001 \cdot 0,501 - 0,5 \cdot 0,995} = \\
 &= \frac{1}{\omega} \cdot \frac{0,5029945}{0,004001} \cong \frac{125,717}{\omega} = 125717[h]
 \end{aligned}$$

c. Tehát a hideg tartalékolt rendszer várható élettartama egy műszeréhez képest kb. 126-szorosára növekedett.

2. Adott egy felújítási Markov-folyamat jelfolyam-gráfja. A nem működő rendszer állapotait piros csomópontokkal jelöltük. A paramétereket a születési dátumából kell generálnia. $\omega = \frac{1}{1945} \left[\frac{1}{h} \right]$, $\alpha = \frac{1}{5} \left[\frac{1}{h} \right]$ és $\mu = \frac{1}{9} \left[\frac{1}{h} \right]$ Mekkora a készenléti tényező

aszimptotikus értéke?



Ad1. állapotér-módszerrel: Az S_i állapotokban tartózkodás valószínűsége függ az időtől. Feltételezzük, hogy ezeknek az $x_i(t)$ függvényeknek létezik az $X_i(s)$ Laplace-transzformáltja. A jelfolyam-gráf alapján felírható egyenletek:

$$X_1(s) = \frac{1}{s} - \frac{\omega}{s} X_1(s) + \frac{\mu}{s} X_3(s)$$

$$X_2(s) = \frac{\omega}{s} X_1(s) - \frac{\alpha}{s} X_2(s)$$

$$X_3(s) = \frac{\alpha}{s} X_2(s) - \frac{\mu}{s} X_3(s)$$

Ha mindhárom egyenletet megszorozzuk az s operátorral és elvégezzük az inverz Laplace-transzformációt, akkor egy differenciálegyenlet-rendszert kapunk, amelynek

két tetszőleges egyenletét kell megtartanunk, mert az állapotjelzők nem függetlenek, hiszen teljes eseményrendszert alkotva érvényes rájuk, hogy

$$\sum_{i=1}^3 x_i(t) = 1$$

$$\dot{x}_1 = -\omega x_1 + \mu x_3 = -(\mu + \omega)x_1 - \mu x_2 + \mu$$

$$\dot{x}_2 = \omega x_1 - \alpha x_2$$

Állandósult állapotban a deriváltak értéke zérus, tehát egy lineáris algebrai egyenletrendszert kapunk, amelynek x_1 megoldása maga a keresett aszimptotikus készenléti tényező.

$$\begin{cases} (\mu + \omega)x_1 + \mu x_2 = \mu \\ \omega x_1 - \alpha x_2 = 0 \end{cases}$$

$$\begin{cases} \alpha(\mu + \omega)x_1 + \alpha\mu x_2 = \alpha\mu \\ \mu\omega x_1 - \alpha\mu x_2 = 0 \end{cases}$$

$$[\alpha(\mu + \omega) + \mu\omega]x_1 = \alpha\mu$$

$$K = x_1 = \frac{\alpha\mu}{\alpha(\mu + \omega) + \mu\omega} = \frac{\frac{1}{5} \cdot \frac{1}{9}}{\frac{1}{5} \left(\frac{1}{9} + \frac{1}{1945} \right) + \frac{1}{9} \cdot \frac{1}{1945}} = \frac{1945}{1945 + 9 + 5} =$$

$$= \frac{1945}{1959} = 0,992853496682 \approx 0,993$$

Ad2. Jelfolyam-gráf kapcsolási módszerrel:

A visszacsatolásnál felnyitott kör átviteli függvénye: $Y(s) = \frac{\omega}{s + \omega} \cdot \frac{\alpha}{s + \alpha} \cdot \frac{\mu}{s + \mu}$.

Az első állapotban tartózkodás valószínűségének Laplace-transzformáltja:

$$X_1(s) = \frac{s}{s + \omega} \cdot \frac{1}{1 - Y(s)} = \frac{(s + \alpha)(s + \mu)}{(s + \alpha)(s + \mu)(s + \omega) - \alpha\mu\omega} =$$

$$= \frac{(s + \alpha)(s + \mu)}{s[s^2 + (\alpha + \mu + \omega)s + \alpha(\mu + \omega) + \mu\omega]}$$

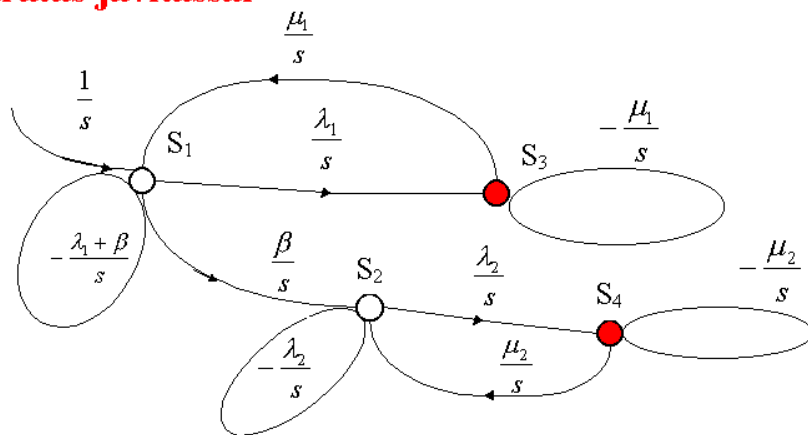
E valószínűségi időfüggvény végtelenben vett határértéke a Laplace-transzformáció végérték-tételével:

$$K = \lim_{t \rightarrow \infty} x_1(t) = \lim_{s \rightarrow 0} sX_1(s) = \lim_{s \rightarrow 0} \frac{(s + \alpha)(s + \mu)}{s^2 + (\alpha + \mu + \omega)s + \alpha(\mu + \omega) + \mu\omega} =$$

$$= \frac{\alpha\mu}{\alpha(\mu + \omega) + \mu\omega}$$

3. Adott egy Markov-folyamat jelfolyam-gráfja. A nem működő rendszer állapotait piros csomópontokkal jelöltük. Mekkora a készenléti tényező aszimptotikus értéke, ha a paraméterek: $\beta = \frac{1}{20h}$, $\lambda_1 = \frac{1}{50h}$, $\lambda_2 = \frac{1}{1000h}$, $\mu_1 = \frac{2}{h}$ és $\mu_2 = \frac{1}{4h}$?

Bejáratás javítással



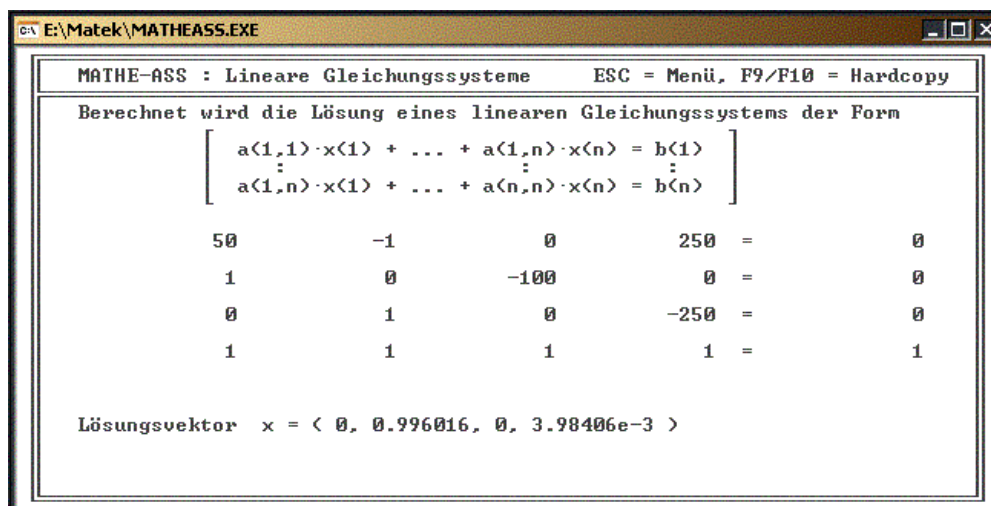
Kidolgozás:

$$\begin{cases} \dot{x}_1 = -(\beta + \lambda_1)x_1 + \mu_1 x_3 \\ \dot{x}_2 = \beta x_1 - \lambda_2 x_2 + \mu_2 x_4 \\ \dot{x}_3 = \lambda_1 x_1 - \mu_1 x_3 \\ \dot{x}_4 = \lambda_2 x_2 - \mu_2 x_4 \end{cases}$$

Állandósult állapotban a deriváltak értéke zérus:

$$\begin{cases} \beta x_1 - \lambda_2 x_2 + \mu_2 x_4 = 0 \\ \lambda_1 x_1 - \mu_1 x_3 = 0 \\ \lambda_2 x_2 - \mu_2 x_4 = 0 \\ x_1 + x_2 + x_3 + x_4 = 1 \end{cases}$$

Az utolsó egyenlet a teljes eseményrendszerre érvényes. Ezzel lecserélve, pl. az első homogén egyenletet, egyértelmű megoldást szolgáltató inhomogén, lineáris, algebrai egyenletrendszert kaptunk, amely a megadott paraméterekkel, de 1000-rel, illetve 100-zal beszorozva a megoldó program képernyőjén látható mátrixos formában:



A keresett aszimptotikus készenléti tényező a működő állapotokban, végül a bejártott állapotban tartózkodás valószínűsége: $K = \lim_{t \rightarrow \infty} K(t) = \lim_{t \rightarrow \infty} x_2(t) \cong 0,996$.

A differenciálegyenletet akkor célszerű megoldani, ha valaki a $K(t)=x_1(t)+ x_2(t)$ függvényre is kíváncsi. A második differenciálegyenlet helyére téve az algebrai egyenletet, a feladatot számítógéppel, konkrétan blokkorientált szimulációval megoldó TUTSIM modell:

MODEL:				
1.0000	1 INT	7	-8	;x1
	2 SUM	9	-1	-3 ;x2
		-4		
0.0000	3 INT	6	-7	;x3
0.0000	4 INT	5	-10	;x4
0.0010000	5 GAI	2		;x2/1000
0.0200000	6 GAI	1		;x1/50
2.0000	7 GAI	3		;2x3
0.0700000	8 GAI	1		;0.07x1
1.0000	9 CON			;1
0.2500000	10 GAI	4		;x4/4
	11 SUM	1	2	;K

A grafikonon felülről lefelé az $x_3(t)$, $x_4(t)$ és $K(t)$ függvény látható:

A számszerű futtatás értéktáblázatának vége:

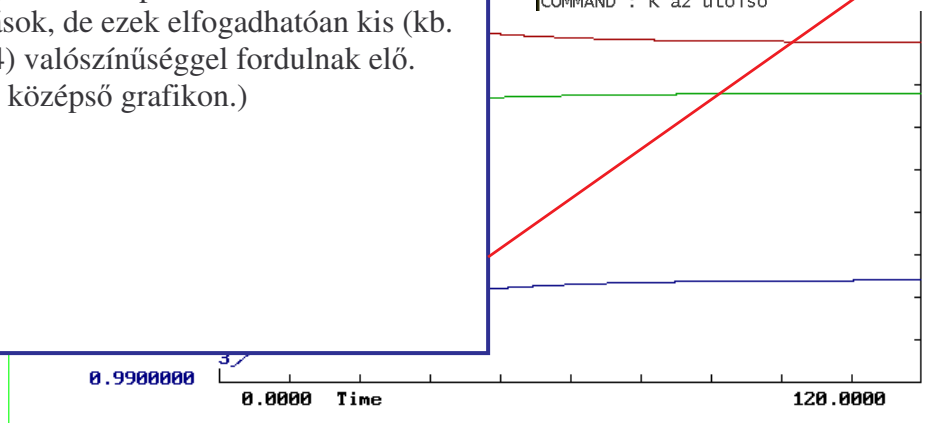
Az alsó (3-mal jelölt) grafikonon látszik, hogy $K(t)$ kezdetben az 1-ről lecsökken, majd felkúszik állandósult értékére, amint a bejáratás teljesen befejeződött.

A felső grafikon éppen azt mutatja, hogy a bejáratáskor fellépő, de gyorsan elhárítható üzemzavar miatti leállás valószínűsége gyorsan eléri maximumát, majd onnan lassan, de biztosan lecsökken egészen nullára.

A bejáratott állapotban is vannak leállások, de ezek elfogadhatóan kis (kb. 0,004) valószínűséggel fordulnak elő. (Ez a középső grafikon.)

294.7500	4.690E-09	0.0039840	0.9960160
295.0000	4.633E-09	0.0039840	0.9960160
295.2500	4.576E-09	0.0039840	0.9960160
295.5000	4.520E-09	0.0039840	0.9960160
295.7500	4.464E-09	0.0039840	0.9960160
296.0000	4.409E-09	0.0039840	0.9960160
296.2500	4.355E-09	0.0039840	0.9960160
296.5000	4.301E-09	0.0039840	0.9960160
296.7500	4.248E-09	0.0039840	0.9960160
297.0000	4.196E-09	0.0039840	0.9960160
297.2500	4.145E-09	0.0039840	0.9960160
297.5000	4.094E-09	0.0039840	0.9960160
297.7500	4.043E-09	0.0039840	0.9960160
298.0000	3.994E-09	0.0039840	0.9960160
298.2500	3.944E-09	0.0039840	0.9960160
298.5000	3.896E-09	0.0039840	0.9960160
298.7500	3.848E-09	0.0039840	0.9960160
299.0000	3.801E-09	0.0039840	0.9960160
299.2500	3.754E-09	0.0039840	0.9960160
299.5000	3.708E-09	0.0039840	0.9960160
299.7500	3.662E-09	0.0039840	0.9960160
300.0000	3.617E-09	0.0039840	0.9960160

COMMAND : K az utolsó



```

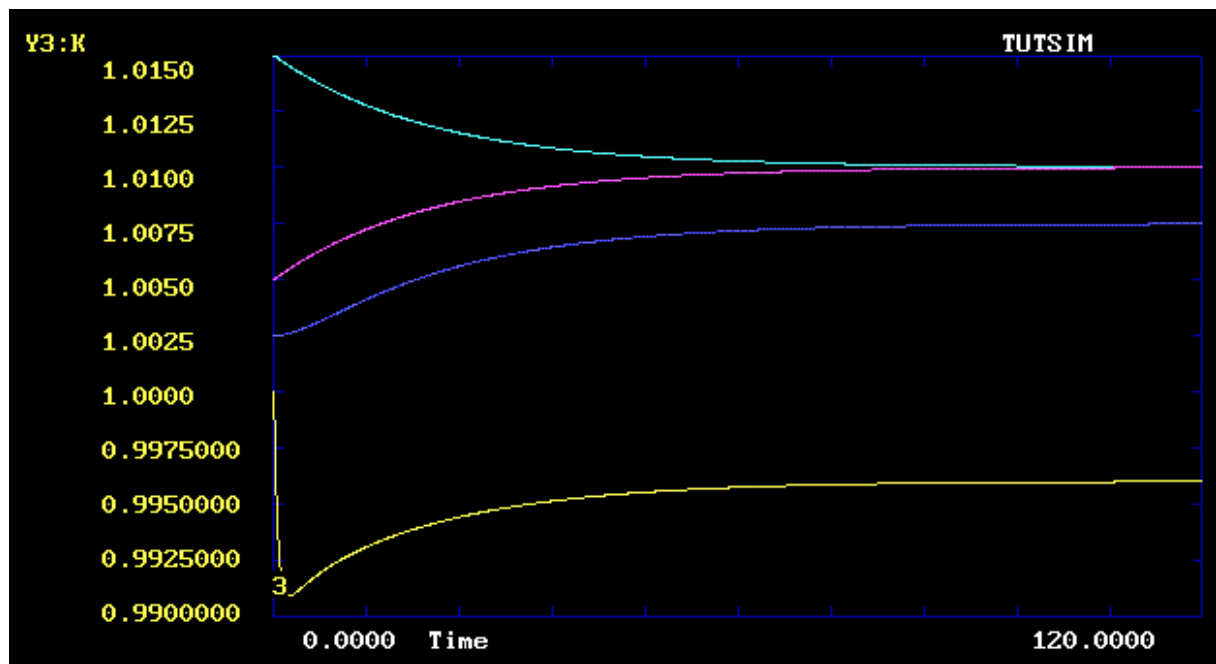
C:\ TUTSIM.EXE
Horz: 0 , 0.0000 , 120.0000 ; Time
Y1: 1 , -4.0000 , 1.0000 ; x1
Y2: 2 , -3.0000 , 2.0000 ; x2
Y3: 10 , 0.9900000 , 1.0150 ; K
Y4: 4 , -0.0100000 , 0.0100000 ; x4

This PERSONAL TUTSIM is not licensed for commercial use by corporations.
Use "EE" command for other license restrictions, including classroom use.

MODEL:
1.0000 1 INT -5 6 -7 ;x1
0.0000 2 INT 7 -8 9 ;x2
0.0000 3 INT 5 -6 ;x3
0.0000 4 INT 8 -9 ;x4
0.0200000 5 GAI 1 ;lam1*x1
2.0000 6 GAI 3 ;mu1*x3
0.0500000 7 GAI 1 ;beta*x1
0.0010000 8 GAI 2 ;lam2*x2
0.2500000 9 GAI 4 ;mu2*x4
10 SUM 1 2 ;K

COMMAND :

```



C:\ TUTSIM.EXE				
294.7500	457.486E-09	0.9960140	0.9960150	0.0039840
295.0000	451.861E-09	0.9960140	0.9960150	0.0039840
295.2500	446.305E-09	0.9960140	0.9960150	0.0039840
295.5000	440.817E-09	0.9960140	0.9960150	0.0039840
295.7500	435.397E-09	0.9960140	0.9960150	0.0039840
296.0000	430.043E-09	0.9960140	0.9960150	0.0039840
296.2500	424.755E-09	0.9960140	0.9960150	0.0039840
296.5000	419.532E-09	0.9960140	0.9960150	0.0039840
296.7500	414.374E-09	0.9960140	0.9960150	0.0039840
297.0000	409.279E-09	0.9960140	0.9960150	0.0039840
297.2500	404.246E-09	0.9960140	0.9960150	0.0039840
297.5000	399.275E-09	0.9960140	0.9960150	0.0039840
297.7500	394.366E-09	0.9960140	0.9960150	0.0039840
298.0000	389.517E-09	0.9960140	0.9960150	0.0039840
298.2500	384.727E-09	0.9960140	0.9960140	0.0039840
298.5000	379.996E-09	0.9960140	0.9960140	0.0039840
298.7500	375.324E-09	0.9960140	0.9960140	0.0039840
299.0000	370.709E-09	0.9960140	0.9960140	0.0039840
299.2500	366.151E-09	0.9960140	0.9960140	0.0039840
299.5000	361.648E-09	0.9960140	0.9960140	0.0039840
299.7500	357.202E-09	0.9960140	0.9960140	0.0039840
300.0000	352.809E-09	0.9960140	0.9960140	0.0039840
COMMAND :				

Ellenőrző kérdések

1. Mit nevezünk Markov-láncnak és mi a Markov-tulajdonság?
2. Melyek a Markov-féle megbízhatósági modell jellemzői?
3. Kibernetikai értelemben az állapotfüggő megbízhatóság-szabályozás modelljének kialakításakor milyen *döntésfolyamatot* kapunk?

10. REKONFIGURÁLHATÓ SZÁMÍTÓGÉPEK (*RECONFIGURABLE COMPUTING*).

Angol nyelvű szakirodalom feldolgozását igénylő tananyagrészt:

1. André DeHon. Comparing Computing Machines. In *Configurable Computing: Technology and Applications*, Proceedings of SPIE 3526, p. 124, November 1998.
http://brass.cs.berkeley.edu/documents/ccmpare_spie98.pdf
2. K. Compton, S. Hauck, "An Introduction to Reconfigurable Computing", *Northwestern University, Dept. of ECE Technical Report*, 1999.
http://www.ece.wisc.edu/~kati/Publications/Compton_ReconfigIntro.pdf
3. K. Compton, S. Hauck, "Reconfigurable Computing: A Survey of Systems and Software", *ACM Computing Surveys*, Vol. 34, No. 2. pp. 171-210. June 2002.
http://www.ece.wisc.edu/~kati/Publications/Compton_ReconfigSurvey.pdf
4. R. Tessier and W. Burleson, Reconfigurable Computing and Digital Signal Processing: A Survey, in *Journal of VLSI Signal Processing*, May/June 2001, pp. 7-27.
<http://www.ecs.umass.edu/ece/tessier/jvsp00.pdf>
5. R. Hartenstein (embedded tutorial): A Decade of Research on Reconfigurable Architectures – a Visionary Retrospective; Proc. International Conference on Design Automation and Testing in Europe 2001 (DATE 2001), Exhibit and Congress Center, Munich, Germany, March 2001.
<http://xputers.informatik.uni-kl.de/papers/paper111.pdf>
6. André DeHon, John Wawrzynek. Reconfigurable Computing: What, Why, and Design Automation Requirements? In *Proceedings of the 1999 Design Automation Conference*, pages 610--615, June 1999.
http://brass.cs.berkeley.edu/documents/rc_dac99.pdf

KÉRDÉSEK

- 10.1 Mi a rekonfigurálható számítógépek elvi gondolata, és mi nyújtja a megvalósításhoz a technológiai háttér?
- 10.2. Milyen az FPGA felépítése és mi a LUT táblázatok szerepe?
- 10.3. Mutassa be, hogyan történik az utasításblokkok áthelyezése a hardverbe!
- 10.4. Mutasson példákat a rekonfigurálható processzorok alkalmazására!