



BUDAPESTI MŰSZAKI ÉS GAZDASÁGTUDOMÁNYI EGYETEM
VILLAMOSMÉRNÖKI ÉS INFORMATIKAI KAR
MÉRÉSTECHNIKA ÉS INFORMÁCIÓS RENDSZEREK TANSZÉK

Digitális technika

VIMIAA01

Fehér Béla
BME MIT

Processzor utasítás rendszerek

- A processzorok utasításkészlete az alábbi utasítás csoportokra osztható:
 - Adatmozgató (MOV, máshol LOAD/STORE)
 - Aritmetikai (ADD, SUB, CMP, MUL, DIV)
 - Logikai (AND, OR, XOR, CPL, TST)
 - Léptető (SLx, /SRx, RR/RL, ASR és multibit)
 - Vezérlő (JMP, CJP, JAL/CALL, RTN)
 - Egyéb (NOP, EI, DI, HALT)
- Jellemző RISC utasításkészlet: 50 – 150 utasítás
- Érdekesség: OISC: One Instruction Set Computer
 - Összetett utasítás művelet, test és adatmozgatás

Processzor utasítás rendszerek

- **A RISC processzorok utasítás formátuma**
 - **Általában fix hosszúságú, 16/32 bit**
 - **Felépítése 2R vagy 3R típusú (utóbbi a 32 bitnél)**
 - **Kevés típusú utasítás formátum (2-5 maximum)**
 - **Egy formátumon belül minden utasítás azonosan használható**
 - **A bitmezők kiosztása rögzített**
 - **Egyszerű dekódolás, sok érték közvetlenül használható**
 - **Pipe-line végrehajtás könnyen kialakítható**
 - **Az utasítások végrehajtási ideje azonos (1 órajel)**

Processzor utasítás rendszerek

- **A RISC processzorok címzési módjai**
 - A címzési mód utasításban hivatkozott adatelem, érték elérési módját jelenti
- **A címzési módok jelentősen befolyásolják az utasítás ill. programvégrehajtás hatékonyságát**
- **Különböző címzési módok léteznek a program memória és az adatmemória elérésére**
 - **Eltérő igények:**
 - Programmemória esetén a cél a következő utasítás címének megadása, ez ugrásoknál, szubrutinhívásoknál (függvény/eljárás) lényeges
 - Adatmemóriánál sokkal általánosabb igényeket találunk

Processzor utasítás rendszerek

- **Programmémória címzési módjai**

- A programmemóriát a PC (programszámláló) címzi
- Az alapművelet a következő utasítás $PC = PC + 1$
- Az ugrási vagy szubrutinhívás esetén a PC értékét módosítani kell
 - **Abszolút címzés:** A PC minden bitje módosul: A teljes programmemória bármely címe szerepelhet, mint új cím. Kedvező, egyszerűen használható, de sok bitet igényel
 - **Relatív címzés:** A PC bitjeinek csak egy része módosul, az ugrás, szubrutinhívás elérési tartománya korlátozott. A programszervezés bonyolultabb (assembler/fordító feladata), de a programok „lokalitása” miatt ez ritkán okoz komoly problémát. A relatív címzési tartomány $2^8 - 2^{16}$

Processzor utasítás rendszerek

- **Programmémória címzési módjai**
 - Közvetlen cím: Az új cím értéke az utasításban kódolt, ez gyakran relatív címzésre korlátoz
 - A célcím FORDÍTÁS időben meghatározott, futáskor konstans, nem lehet adatfüggő
 - MiniRISC: 16 bit utasítás, 8 bit program cím: lehet abszolút is
 - Regiszter indirekt: Az ugrási címet egy regiszter tartalmazza (ez általában abszolút címzésre is elegendő)
 - A célcím lehet adatfüggő, több utas elágazások, táblázat értékek indexelt olvasásához
 - MiniRISC: van regiszter indirekt címzés, teljes tartományra
 - A szegmensregiszteres címzés egy kevert címzési mód

Processzor utasítás rendszerek

- **Adatmemória címzési módjai, illetve az utasítások operandusainak elérési, specifikációs módjai:**
 - **Közvetlen adat**
 - **Regiszter adat**
 - **Regiszter (indirekt) címzésű adat**
 - **Regiszter + közvetlen offset című adat**
 - **Regiszter címzésű indexelt adat**
 - **Regiszter címzésű post-inkremens/pre-dekremens adat**
- **Egy-egy utasításkészlet nem mindegyiket tartalmazza**
- **Léteznek egyéb speciális (bitfordított, modulo, stb.) címzések is, ezeket nem tárgyaljuk**

Processzor utasítás rendszerek

- **Adatmemória címzési módjai, illetve az utasítások operandusainak elérési, specifikációs módjai:**
 - **Közvetlen adat:** Az adat az utasítás része
 - MiniRISC 16 bit utasítás. 8 bit adat → megoldható
 - Általában pl. 32 bites processzor esetén nem egyszerű: 32 bites utasításban 32 bites adat nem helyezhető el (ut.kód?)
 - Gyakran használunk „kis értékű” konstansokat, ezért a közvetlen adatra egy 8-12-16-20 bitmező áll rendelkezésre + előjel kiterjesztés
 - Vagy: teljes méretű adat: két utasításban: alsó-felső fele, és ekkor a két utasítás végrehajtása nem megszakítható!

Processzor utasítás rendszerek

- **Adatmemória címzési módjai, illetve az utasítások operandusainak elérési, specifikációs módjai:**
 - **Regiszter adat:** Az adat címe az utasítás része
 - MiniRISC 16 bit utasítás: 2 regiszter cím \rightarrow 2 adat
 - Ez a leggyakoribb címzési mód
 - RISC processzorokban univerzális regiszterhasználat, nincsenek specialitások, kivételek
 - **Regiszter (indirekt) címzésű adat:** A regiszter tartalma az adat címe a memóriában
 - A MiniRISC utasításkészlet tartalmazza
 - Hatékony címzés adatvektorok/tömbök eléréséhez

Processzor utasítás rendszerek

- **Adatmemória címzési módjai, illetve az utasítások operandusainak elérési, specifikációs módjai:**
 - **Regiszter + közvetlen offszet adat:** Az adat címe egy regisztertartalom (bázis) és az utasításból származó közvetlen érték (offset) összege
 - **Regiszter címzésű indexelt adat:** Az adat címe két regiszter tartalmából adódik: Az egyik a bázis cím a másik az index érték
 - **Regiszter címzésű post-inkremens/pre-dekremens címzés:** Az adat elérése után/előtt a címregiszter tartalma automatikusan módosul. Hasznos verem (stack) és adatvektor/tömb címzésénél

MiniRISC utasításkészlete

A MiniRISC processzor diasorozat 23 - 37 diája

Processzor belső kommunikációja

- A mikroprocesszoros rendszerek összetett digitális rendszerek, több modulból állnak
- Eddig csak a központi egység felépítését, az un. **CPU struktúráját és működését vizsgáltuk:**
 - Ez két fő részegységből áll: vezérlő és ALU
 - Ezek mindegyike bonyolult részrendszer, de önmagában szinte működésképtelen
 - A vezérlő a működéséhez szükséges utasításokat a **programmemória interfészen** keresztül éri el
 - Az ALU értelmes működésének feltétele az adatmemória és a perifériák elérése. Ezt az **adatmemória interfész** biztosítja

Processzor belső kommunikációja

- A mikroprocesszoros rendszerek adatátviteli interfészeit buszoknak nevezzük
- Sok mikroprocesszor rendelkezik egyedi külső busszal is (lokális busz), alkalmazási céltól függően
- A külső buszok némelyike processzor független, ún. rendszer busz, általános célú alkalmazásokra.
- Ilyen külső rendszerbusz pl.
 - IBM PC ISA (Industry Standard Architecture)
 - PCI busz (Peripheral Component Interface)
 - VMEbus (Versa Module Europa bus)
 - VXI (VMEbus eXtension for Instrumentation)

Processzor belső kommunikációja

- A mikroprocesszoros rendszerek belső adatátviteli buszai hasonlóak a külső buszokhoz
- Egy lényeges különbség: Szétválasztott DATA_IN és DATA_OUT adatvonalak vannak.
- A belső buszok között is van processzor független, általános célú rendszer busz, tetszőleges alkalmazásokra. Ez nagyban megkönnyíti a modern SoC (System on Chip) rendszerek tervezését.
- Ilyen belső rendszerbusz pl.
 - IBM Core Connect
 - ARM AMBA (Advanced Microcontr. Bus Arch.)
 - ARM AXI (Advanced eXtensible Interface)

Processzor belső kommunikációja

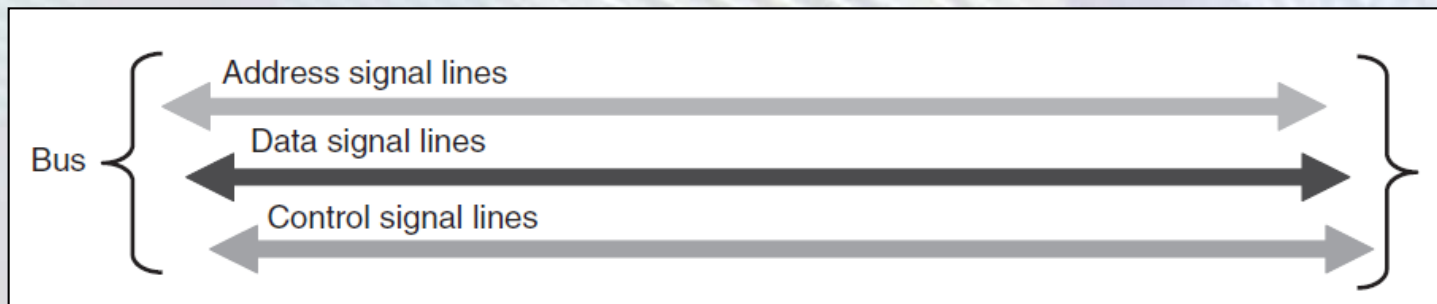
- A felsoroltak mind **PÁRHUZAMOS** buszok
- A mai nagysebességű adatátviteli interfészek gyakran áramköri okokból **SOROS** adatátviteli protokollt használnak. Ezek pont-pont kapcsolatot valósítanak meg, de megszokásból ezeket is buszoknak nevezik.
- Ilyen pont-pont kapcsolatok pl.:
 - **USB (Universal Serial Bus)**
 - **FireWire (IEEE 1394)**
 - **PCIe (PCI Express)**
 - **InfiniBand**
 - **HyperTransport**
 - **QuickPath Interconnect**

Processzor belső kommunikációja

- **A továbbiakban a PÁRHUZAMOS buszokkal foglalkozunk**
- **A busz jelvezetékek együttese, amelyekre áramköri, elektromos és időzítési előírások vonatkoznak**
 - Gyakran mechanikai is pl. ISA/PCI/VME csatlakozó
 - Az áramköri előírások specifikálják a jelszintet, a meghajtó képességet, a megengedett terhelést, a maximális túllövést, stb....
 - Az időzítési előírások definiálják a működési sebességet, a jelváltási időt, az adatátviteli buszciklusok időbeli lefolyását
 - A buszok komplex rendszerek, mi csak egy egyszerű verzióval foglalkozunk

A mikroprocesszoros busz

- **A busz részei:**
 - **Címbusz ADDR[n:0]**
 - **Adatbusz DATA[m:0],**
belső busznál külön **D_IN[m:0], D_OUT[m:0]**
 - **Vezérlő busz (sok egyedi jel összessége):**
 - **Rendszerjelek: CLK, RST,**
 - **Arbitrációs jelek: BUSREQ, BUSACK,**
 - **Írányvezérlő jelek: READ, WRITE,**
 - **Átvitelvezérlő jelek: FRAME, TS, TACK, AS, DS,**
 - **Megszakítás vezérlő jelek: IRQi, IACK,**

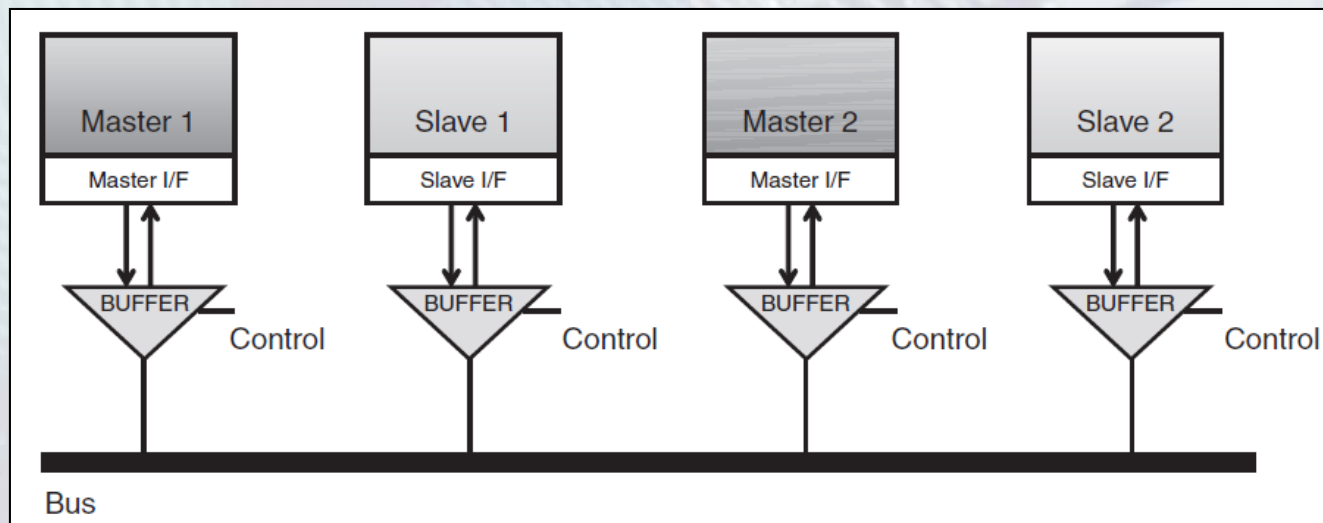


A mikroprocesszoros busz

- **A busz használat szabályai:**
 - A buszra kapcsolódó egységek között megkülönböztetünk **MASTER** és **SLAVE** egységeket
 - **MASTER:** Vezérelheti a buszt, meghatározza a cím/adat és egyéb paramétereket, kezdeményezi az átvitelt és vezérli a működést
 - **SLAVE:** Figyeli a buszt, felismeri/dekódolja a címet és a parancsokat, azonosítás esetén válaszol a kérésre az adat fogadásával vagy kiadásával

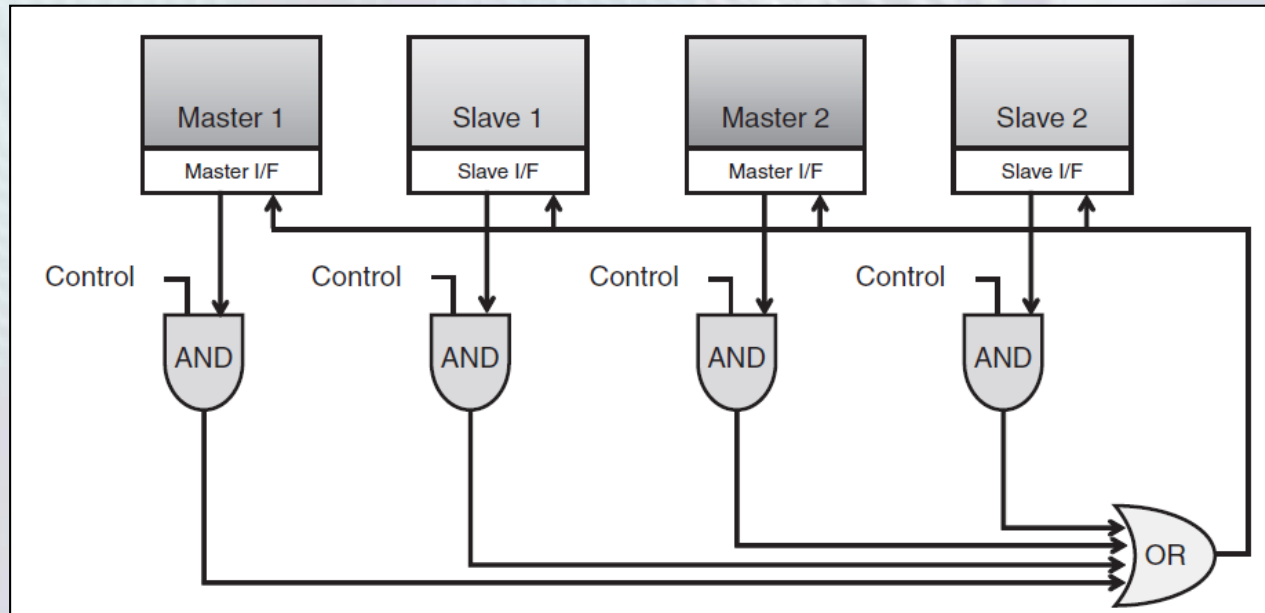
A mikroprocesszoros busz

- Buszhasználat külső buszon:
- Közös vonalak, 3 állapotú, HiZ, nagy impedanciás meghajtókkal
- Egy időben csak egyetlen jel lehet aktív
- A vezérlő/engedélyező jelek 1-az-N-ből kódolásúak



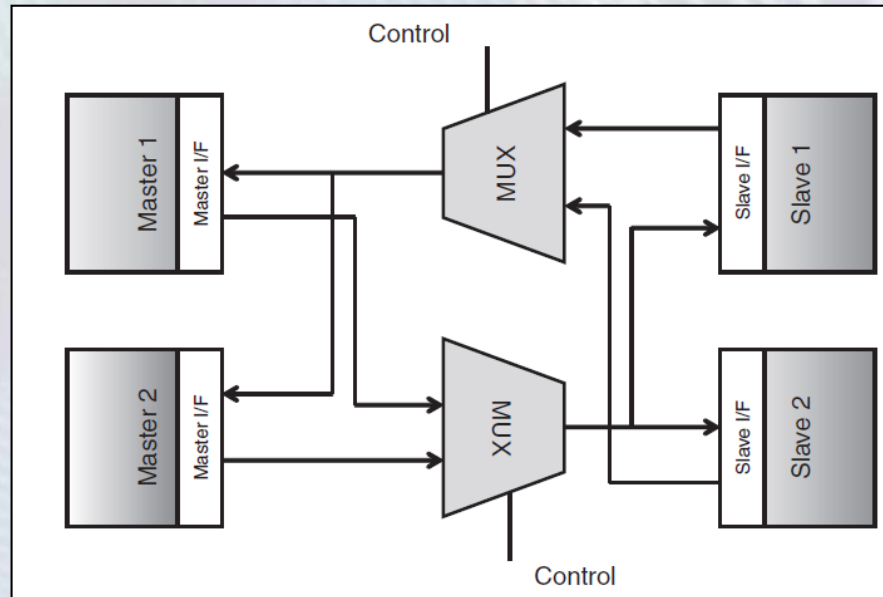
A mikroprocesszoros busz

- Buszhasználat belső buszon:
 - Az adatvonalak meghajtása AND-OR hálózaton (elosztott multiplexer) keresztül
 - A megoldás kizárja a több forrás kimeneteinek áramköri összekapcsolását



A mikroprocesszoros busz

- Buszhasználat belső buszhálózaton:
 - Szétválasztott DATA_IN (SLAVE → MASTER) és DATA_OUT (MASTER → SLAVE)
 - A MUX alapú meghajtás kizárja a több forrás kimenetének áramköri összekapcsolását

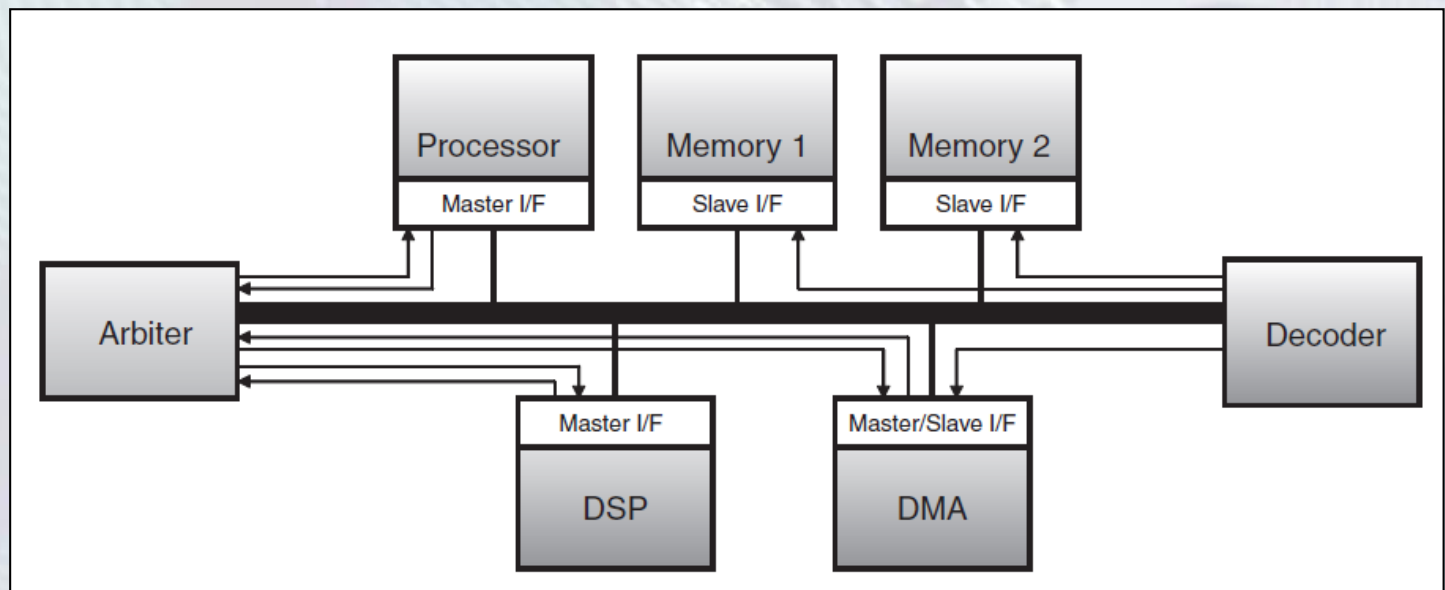


A buszhasználat szabályai

- **Egy buszon lehet több MASTER és SLAVE is**
 - **Több SLAVE nem gond**
 - **Több MASTER : Arbitráció a buszhasználat jogáért**
- **Az arbitrációban csak a MASTER funkciójú egységek vesznek részt**
- **Az arbitráció a buszhasználat jogáért történik**
- **Az arbitráció dönti el, hogy versenyhelyzetben ki jogosult a busz használatára.**
- **Többfajta algoritmus létezik az arbitrációra:**
 - **Fix prioritás, körbenforgó, utolsó egység nagyobb prioritású stb.**

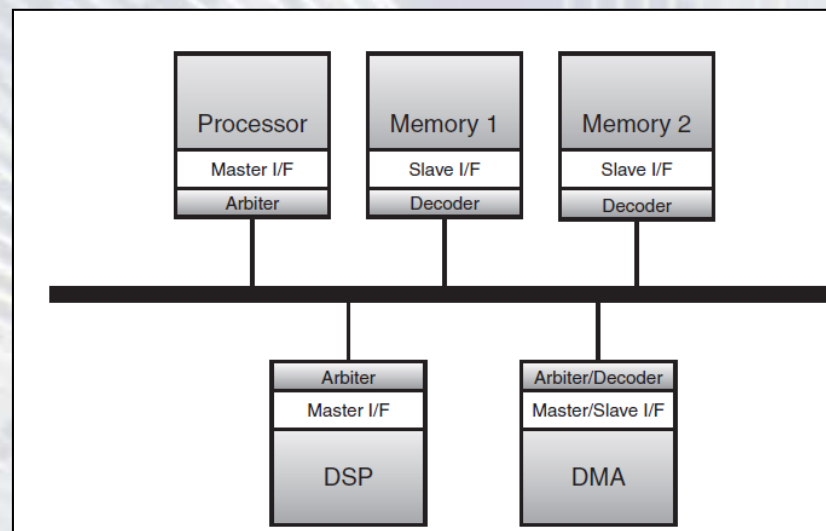
A buszhasználat szabályai

- A buszhasználati feltételek eldöntése lehet:
- **Központi:**
 - Az arbiter értékeli ki a kéréseket és jelöli ki a következő **MASTER** egységet, aki indíthat egy adatátvitelt
 - A dekóder értelmezi a címet/vezérlést és jelöli ki a **SLAVE** egységet, aki válaszol a kérésre



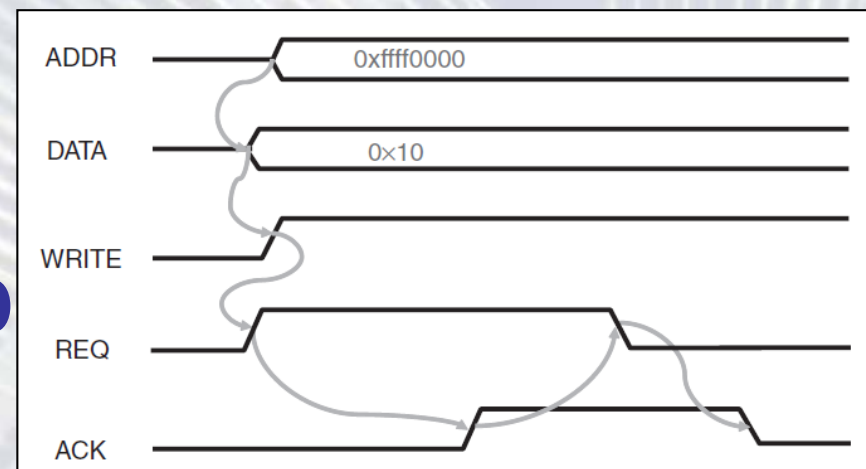
A buszhasználat szabályai

- A buszhasználati feltételek eldöntése lehet:
- **Elosztott:**
 - Minden MASTER tartalmaz logikát az arbitrációhoz és így dől el, ki lehet a következő MASTER egység, aki indíthat egy adatátvitelt
 - Minden SLAVE egység tartalmaz dekóder áramkört, ami értelmezi a címet/vezérlést és engedélyezi az egységet, hogy a kérésre válaszoljon
 - Helyes tervezés esetén csak egy MASTER és egy SLAVE egység lehet egyidőben aktív



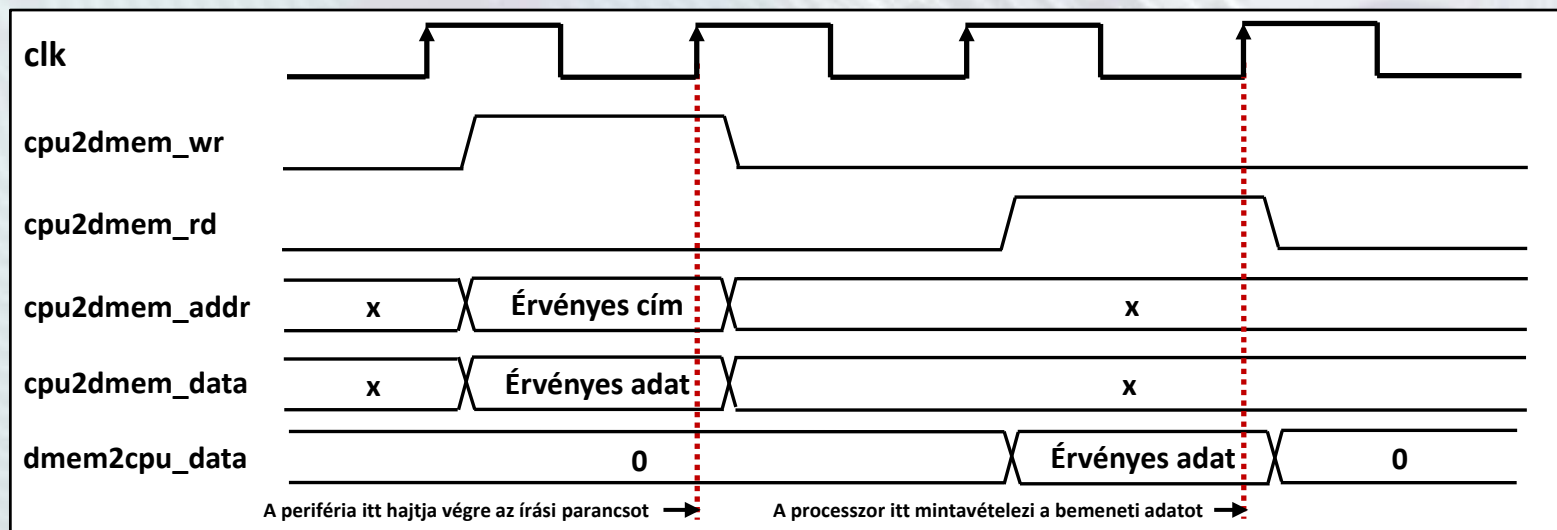
A busz adatátviteli ütemezése

- Az adatátvitel ütemezése aszinkron
- Nincs ütemező órajel
- Az adatátvitelt a „kézfogásos” (hand-shake) szinkronizáció vezérli (REQ → ACK jelek)
- 4 állapot REQ_ACK → 00 – 10 – 11 – 01 - 00
- Ha az előző átvitelnek vége (ACK =0), akkor
- Átviteli paraméterek beállítása
- REQ → 1, start
- Vár ACK-ra, minden jel stabilan tartva
- Ha ACK megjön, REQ=0
- Vár ACK elengedésére



A busz adatátviteli ütemezése

- Az adatátvitel ütemezése szinkron módon
- Az adatátvitelt a busz órajel ciklusai ütemezik
 - Sokfajta busz ütemezési protokoll létezik, gyakori a címzési és az adat fázis szétválasztása
 - Első ütem: a cím és a vezérlési paraméterek beállítása
 - Második ütem: adatátvitel
 - A MiniRISC busz egyetlen órajel alatt hajtja végre az átvitelt



MiniRISC interfészei

A MiniRISC processzor diasorozat 48 - 54 diája

Digitális technika

10. EA vége