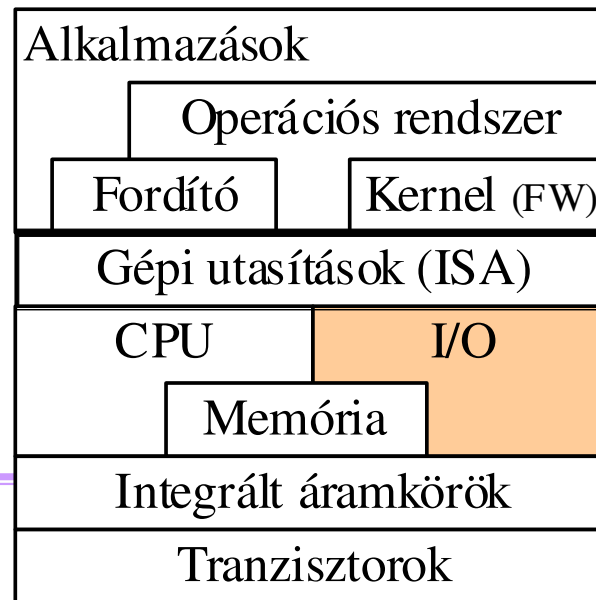


INFORMATIKA I.

BMEVIIIAB08

Számítógép architektúrák *Sín arbitráció*



A sínvezérlési jog eldöntésének mechanizmusa

☐ Statikus

- Valamennyi lehetséges *mester előre meghatározott időzítéssel kapja meg a vezérlés jogát*

☞ /pl.: M1..M4 mester: T1-T2-ben /2T ideig /M1, T3-T6-ban /4T/ M2, T7-T9-ben /3T/ M3, T10-T11-ben /2T/ M4 kapja meg a sínvezérlés jogát, függetlenül az

→ **Túl merev** ⌚👎

Még leginkább szinkron 👍
protokollnál lehet használni

👍 **Egyszerű hw**

👍 **Garantált**

- **válaszidő** 👍

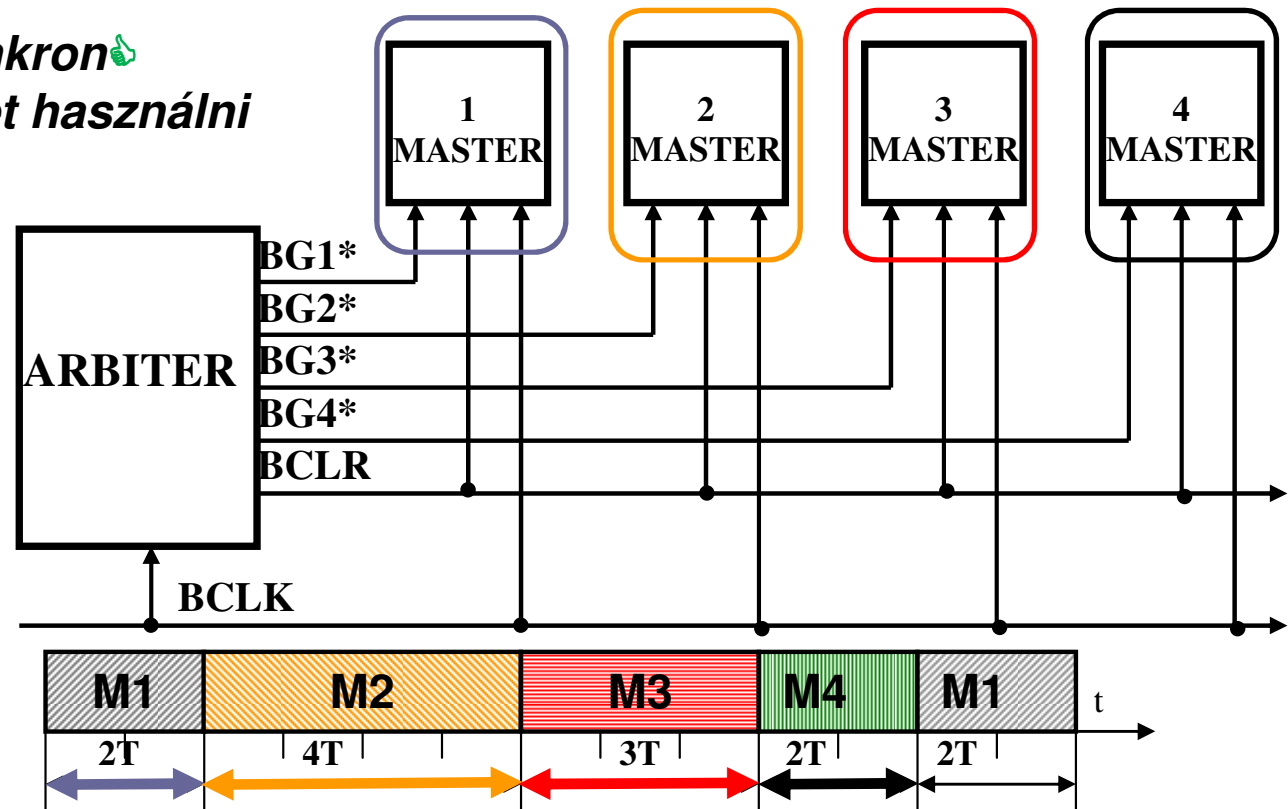
- **busz áteresztő**

kapacitás 👍

mindegyik

mester

számára



- ❑ Dinamikus
 - *Vezérlési jog biztosítása az igények függvényében*
- ✧ -A vezérlési jog odaítélését realizáló mechanizmus a *busz megszerzési és busz elengedési stratégiák megvalósulása*
- ❖ **Busz megszerzési stratégiák/Bus allocation policies/**
- ❑ **Prioritásos /Priority based/**
 - Meghatározott *fontossági sorrend* szerint dönt az egyidejű kérésekről
 - *Statikus* feladat hozzárendelésű processzoroknál lehet előnyös → pl.: I/O alrendszer
- ❑ **Egyenlő esélyű /Fairness/**

Garantálja, hogy *minden mester igénye kiszolgálásra kerül* mielőtt a jelentkezők közül *valaki másodszor is vezérlési jogot kapna* → Megelőzi a kiéheztetést 👍

→ A mesterek prioritása hosszabb időre egyenlő ⏳👎

- ❑ Kombinált , az *előzőek együttes* használata
Pl.: I/O prioritásos és a többi processzor Fairness
- ❖ **Busz elengedési stratégiák /Bus deallocation policies/**
- ❑ Elengedés kéréskor (*Relase on request /ROR/*)
 - A busz mester mindaddig *magánál tartja a vezérlést,*
amíg más nem kéri 🖱️ *pl.: CPU-DMA*
 - 👍 Gyors, újabb igénynél nem kell a sín odaítélésére várni
- ❑ Elengedés ha kész /*Relase when done /RWD//*
 - Ha befejezi az átvitelt, elengedi a vezérlést
 - Mindig újra kell kérni 🖱️
- ❑ Befejezés előtti elengedés /*Pre emption /PRE//*
 - Magasabb prioritású kérés esetén *elengedi a*
vezérlést, mielőtt az összes adatot átvitte volna
 - *Nagy adatblokkok átvitelénél lehet szükség rá* ⌚

□ Központosított (**Centralizált**)

□ Elosztott (**Decentralizált**)

❖ Centralizált hardver mechanizmus

- *Arbiter hardver egy helyen* →

Rendszervezélőben

- *A mesterek kérik a buszvezérlés jogát*

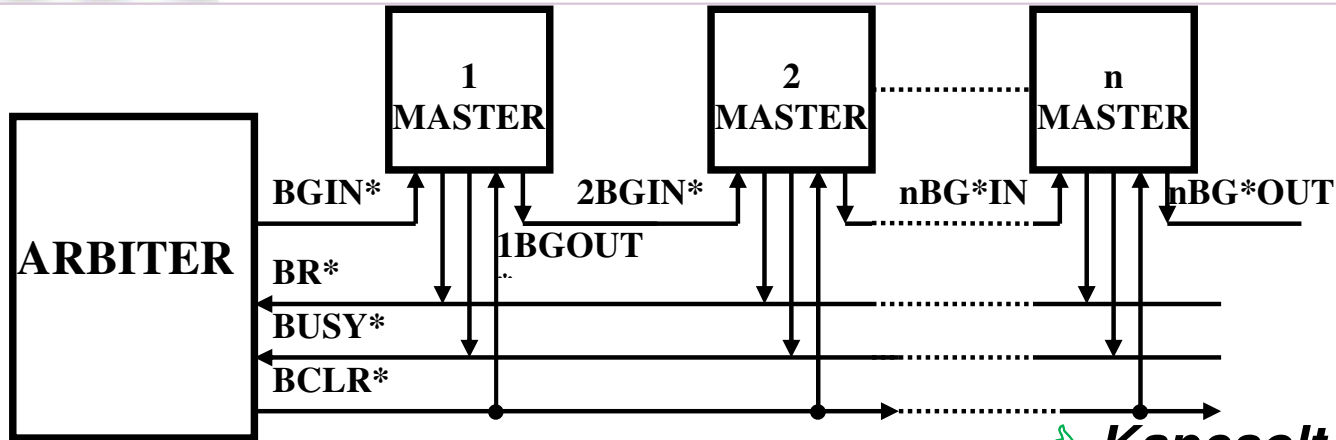
- *Az arbiter dönt*

- *Dönt*

- *Nyugtáz*

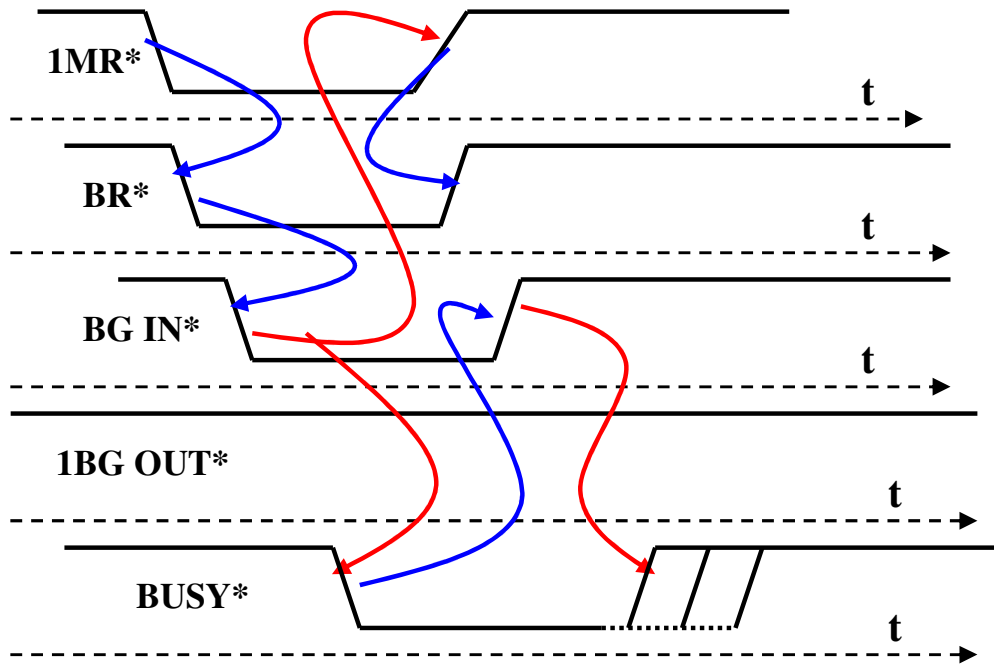
→ *Visszajelzi a jog odaítélését*

◆ Közös kérés-felfűzött válasz hardver mechanizmus



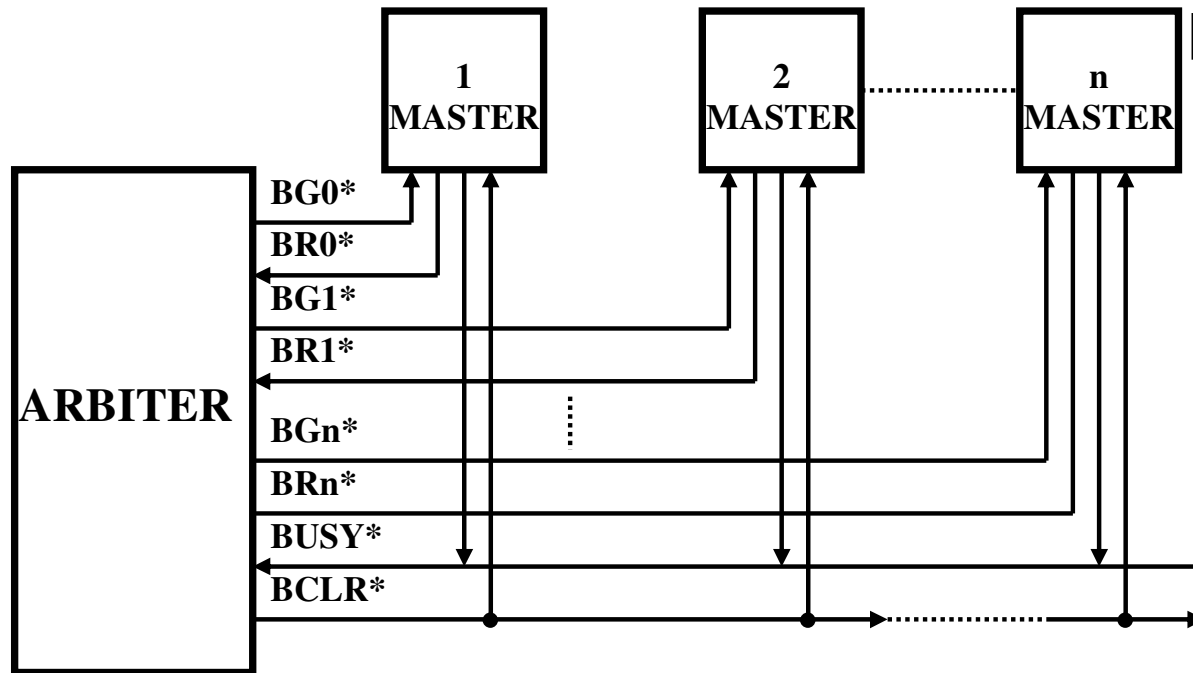
- **Láncolásos felfűzés**
/daisy chain/

👍 **Egyszerű**
👍 **Bővíthető**



- 👍 **Kapcsolt jelek**
- **Kérés-válasz -BUSY* vonal**
- **Hibamentes működés**
- 🗑️ **Merev prioritásos buszmegszerzési stratégia**
- **Az arbiterhez közelebbi mester a magasabb prioritású**
- 🗑️ **Döntés lassú ⌚ lehet**
- 🗑️ **Meghibásodás hatása ❓**

- ◆ **Független kérés-független válasz** hardver mechanizmus
- Minden mester rendelkezik **külön kérés –és válasz vonallal**
- A sín használatát a mesterek a **BUSY*** vonalon jelzik



Előny ↔ Hátrány

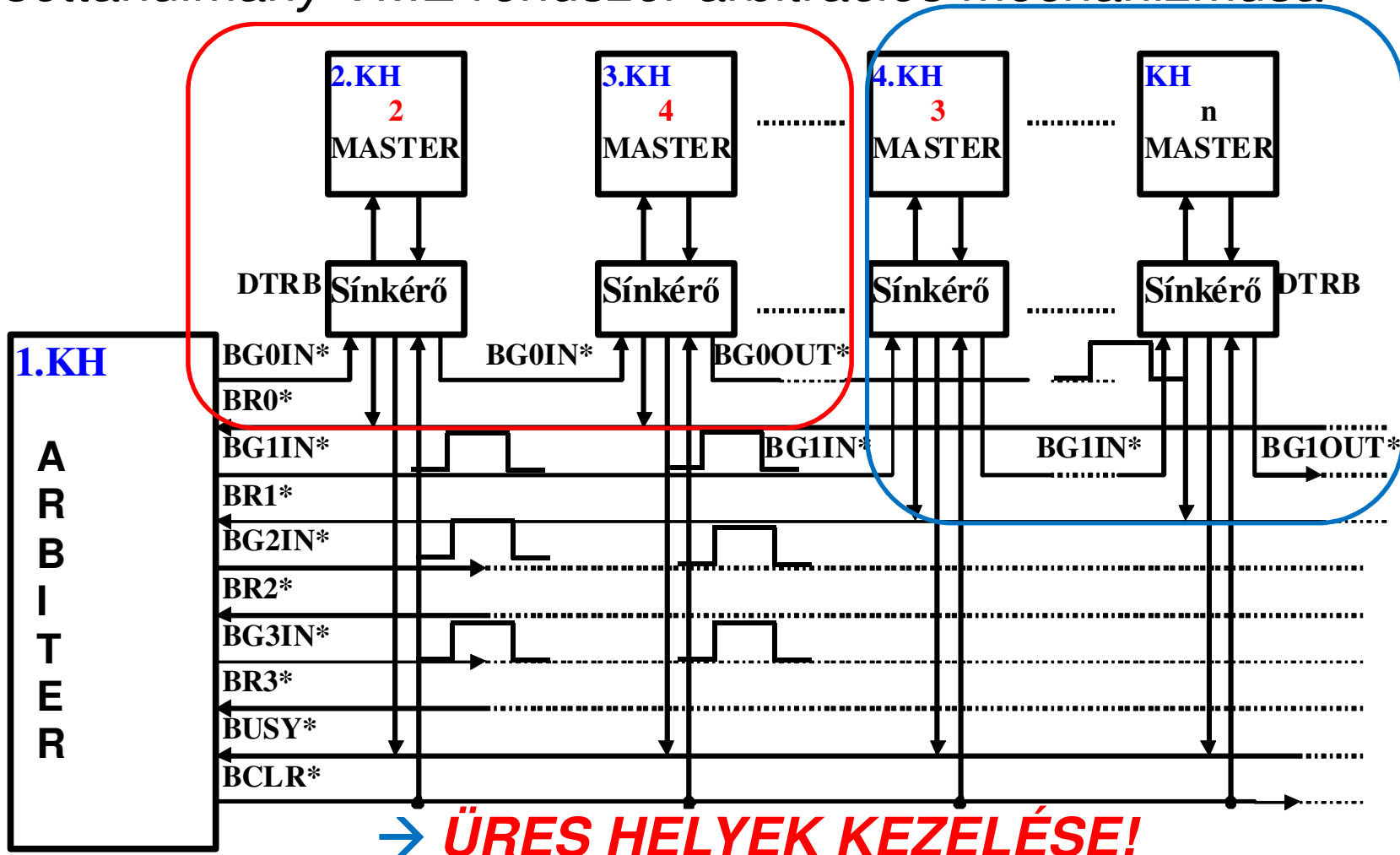
- 👍 **Bármilyen busz megszerzési stratégia alkalmazható**
- 👍 **Gyors működés**
- 👍 **Hibára kevésbé érzékeny**

- 👎 **Sok vonal kell a megvalósításhoz**
- 👎 **Nem bővíthető**

◆ Kombinált mechanizmus

→ A két módszer kombinációja

Esettanulmány VME rendszer arbitrációs mechanizmusa



Realizációk

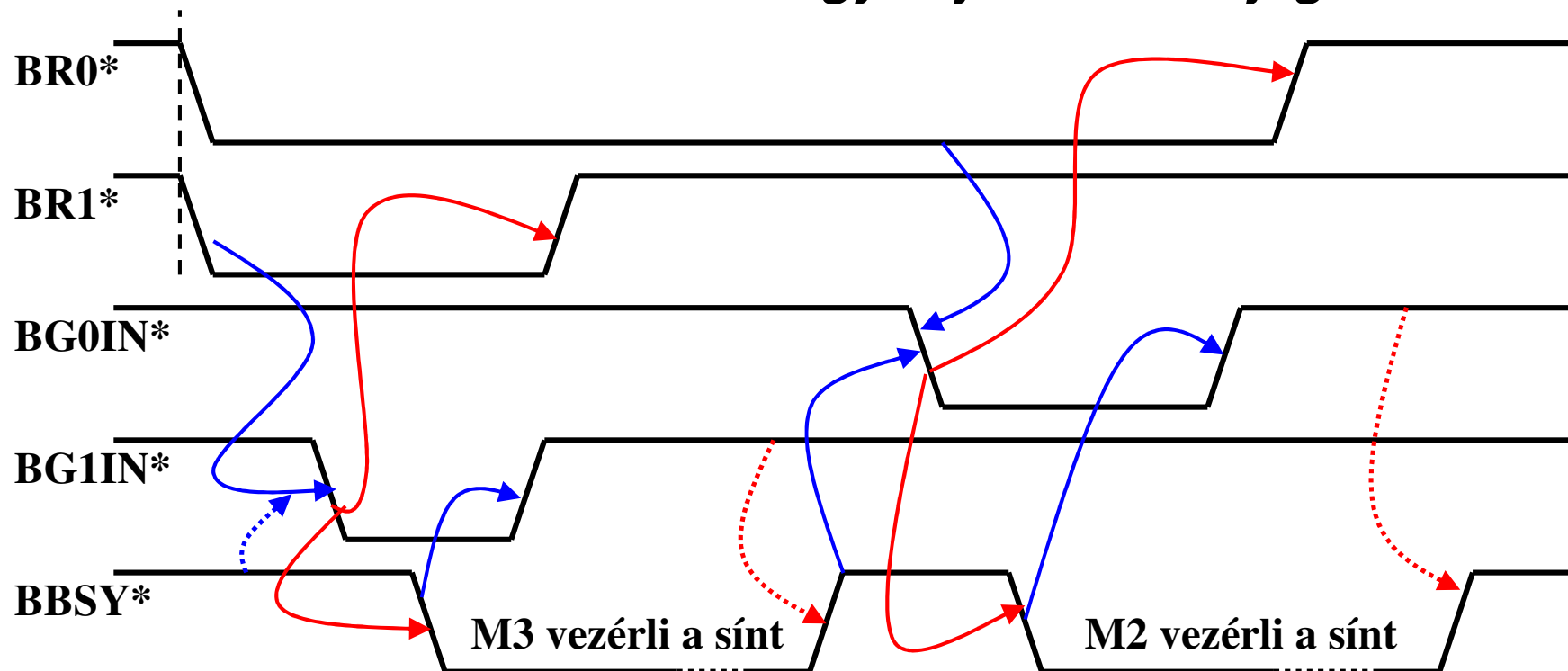
- Egyszintű ($BR3^*$ - $BG3^*$) csak *fix prioritású* lehet
- Többszintű ($BR0^*$ - $BG0^*$.. $BR3^*$ - $BG3^*$)
- *Buszmegszerzési stratégiák*
 - Prioritásos megszerzési stratégia
 - Szintek között és szinten belül felfűzés
 - Egyenlő esélyű → Szintek között forgó prioritású
 - Round-Robin/**RR**/
 - Vegyes → szintek között forgó
 - szinten belül fix prioritás
- *Buszelengedési stratégiák*
 - **RWD**
 - **ROR** → Figyelnie kell a többi kérő vonalat is
 - Mesternek **BCLR***, vagy **BRn*** jelez
 - **PRE** → Mesternek **BCLR***, vagy **BRn*** jelez

Esettanulmány VME rendszer arbitrációs mechanizmusa

Különböző szintű kérések kiszolgálása

→ Kapcsolt jelű protokoll

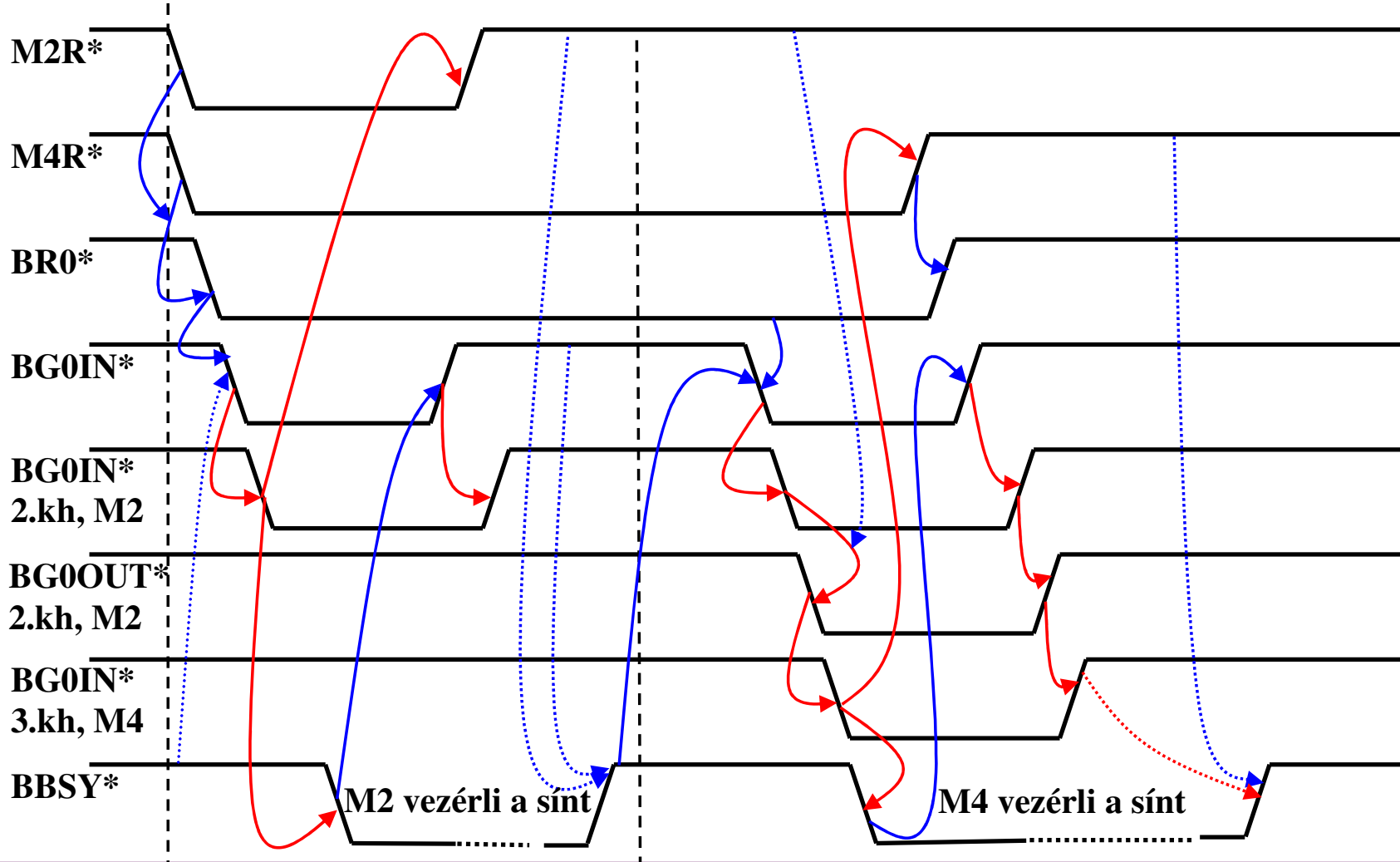
M2 /BR0*/ és M3 /BR1*/ egyidejű vezérlési jog kérése



• Azonos szintű kérések kiszolgálása

→ Kapcsolt jelű protokoll

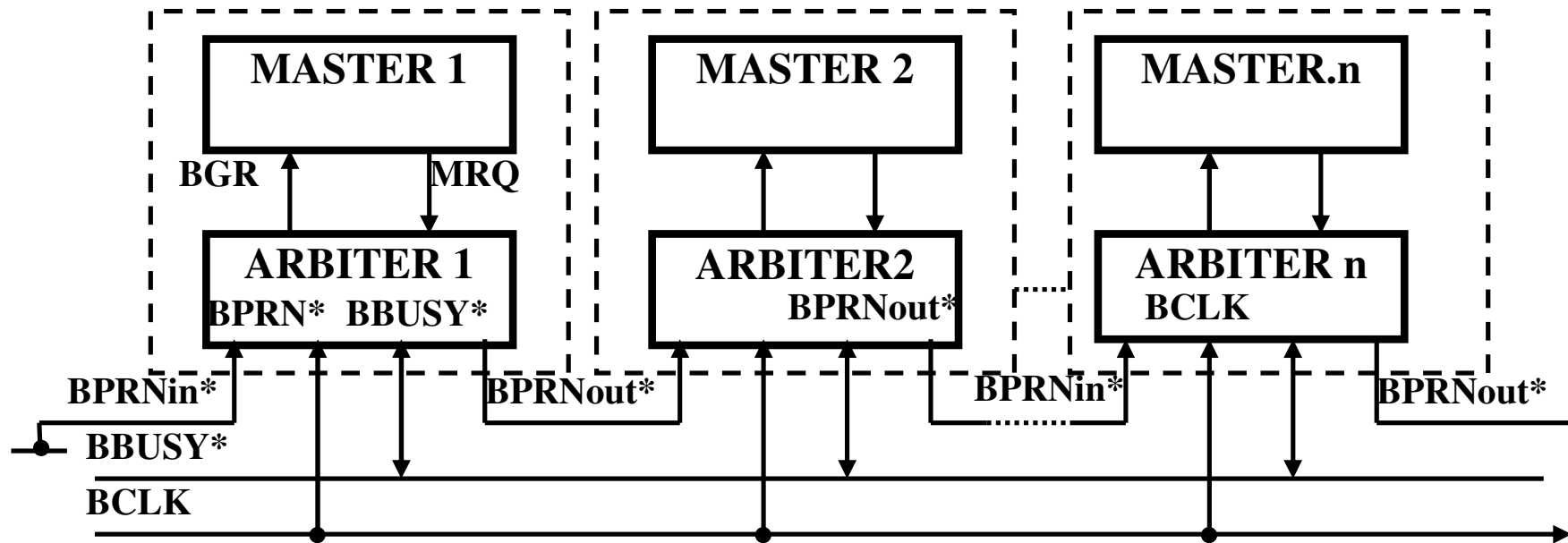
M2/2.KH/ és M4/3.KH/ egyidejű sínvezérlési jog kérése /BR0*/ és kiszolgálása



Elosztott (*decentralizált*) hardver mechanizmus

◆ Soros, elosztott szinkron arbitációs mechanizmus

Láncolós elosztott arbiter



Pl.: MULTIBUS I. max. 4 arbiter → BCLK

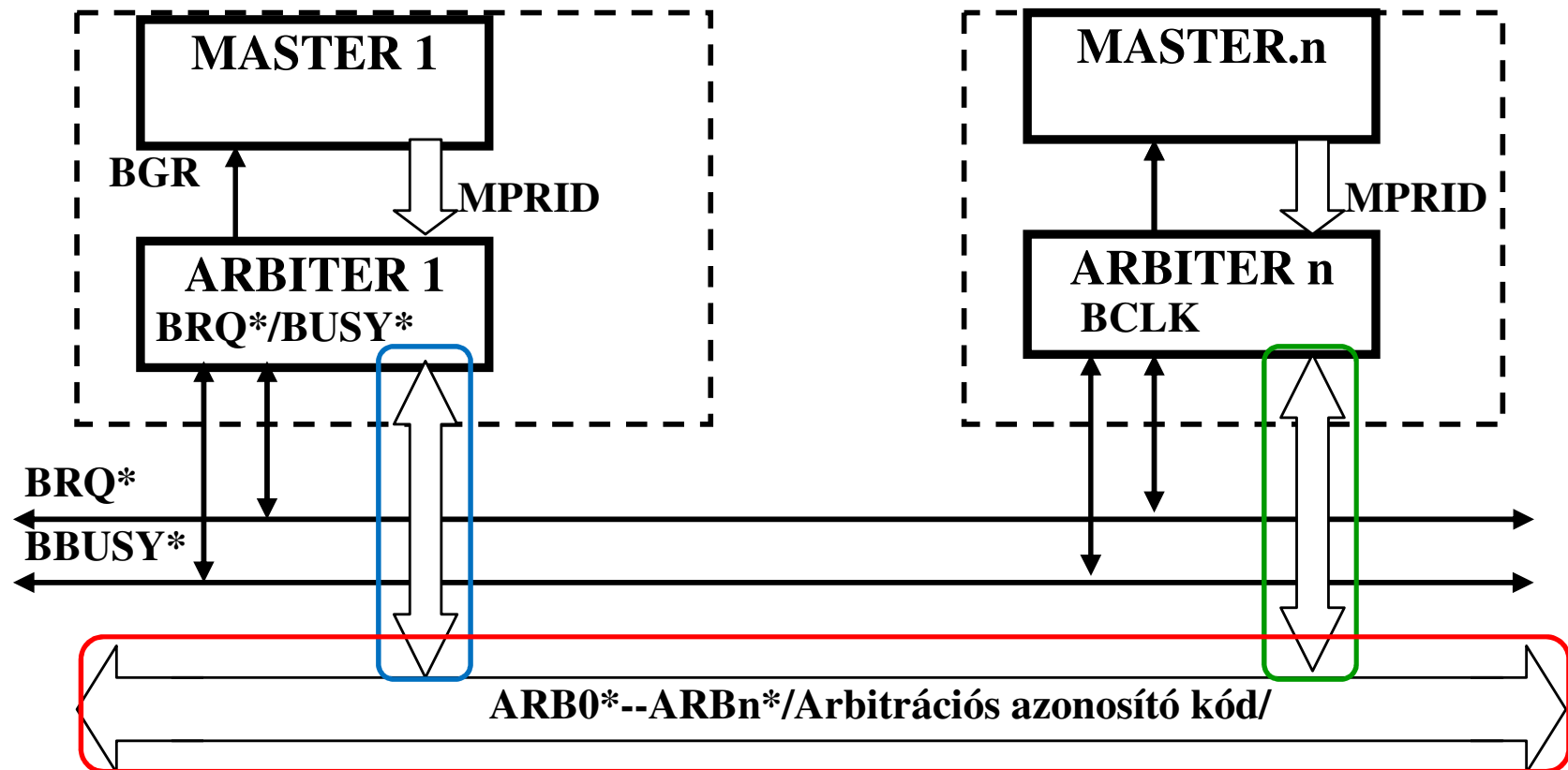
👍 **Egyszerű**

👍 **Könnyen bővíthető**

👎 **Korlátozott az egységek száma (→ Jelterjedési idő → órajel)**

Párhuzamos elosztott arbitrációs mechanizmus

Párhuzamos versenyeztetés mechanizmusú arbiter



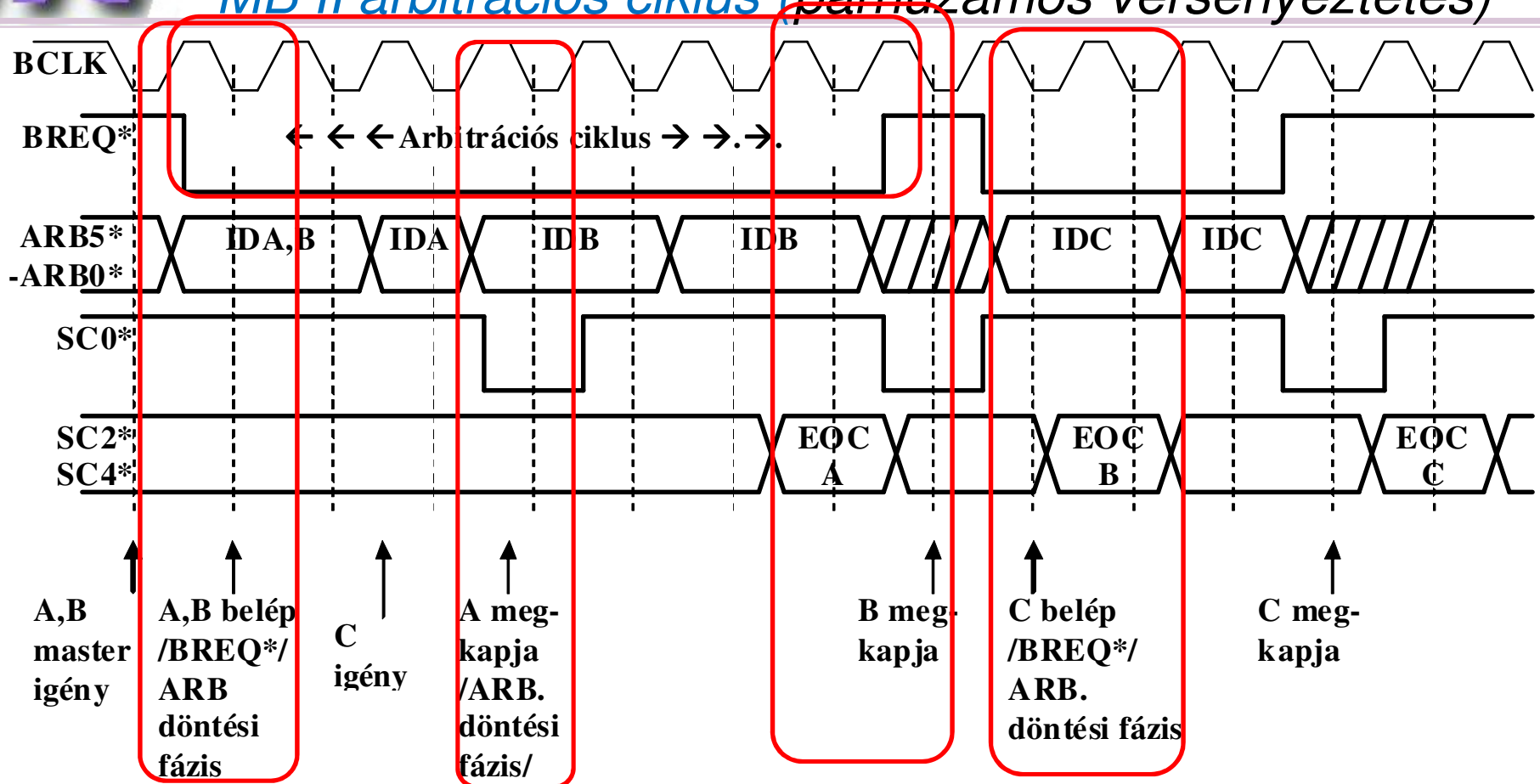
👍 *Rugalmas*

→ *Tetszőleges megszerzési stratégia*

👍 *Mesterek száma közepes → pl.: MB-II. max.20*

Esettanulmány MULTIBUS II

MB II arbitrációs ciklus (párhuzamos versenyeztetés)



Fázisokra bontott működési ciklusok

- MB II arbitrációs ciklus
- Transzfer ciklus
- Átlapolt arbitrációs - és transzfer ciklus

-Párhuzamos versenyeztetés

→ 3 órajel alatt dönt

-Igazságos (egyenlő esélyű) buszmegszerzési stratégia

-ARB5*=0 (L) → Különleges prioritású kérés

-Beléphet az arbitrációs ciklus döntési fázisába

-az a fázis ahol elkezdődik a 3 BCLK-ig tartó döntés

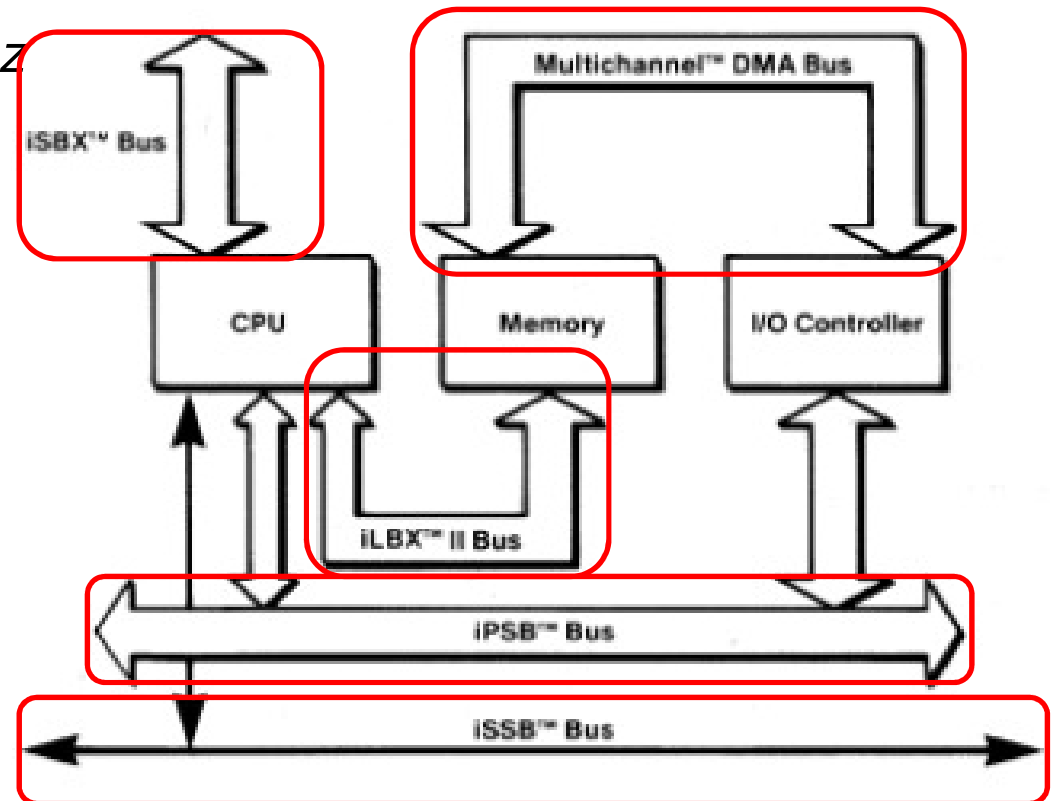
PÉLDA A, B, C egységek → belépnek az arbitrációs ciklusba

1.BCLK	ARB5*	ARB0*	2.BCLK	ARB5*	ARB0*	3.BCLK	ARB5*	ARB0*
A	1 0 0 0	1 0	A	1 0 0 0	1 1	A	1 0 0 0	1 0
B	1 0 0 0	1 1	B	1 0 0 0	1 1	B	1 0 0 0	1 1
C	1 0 1 0	0 0	C	1 0 1 1	1 1	C	1 0 1 1	1 1
SÍN	1 0 0 0	0 0	SÍN	1 0 0 0	1 1	SÍN	1 0 0 0	1 0

→ Sínvezérlési jog kiszolgálási sorrend A, B, C

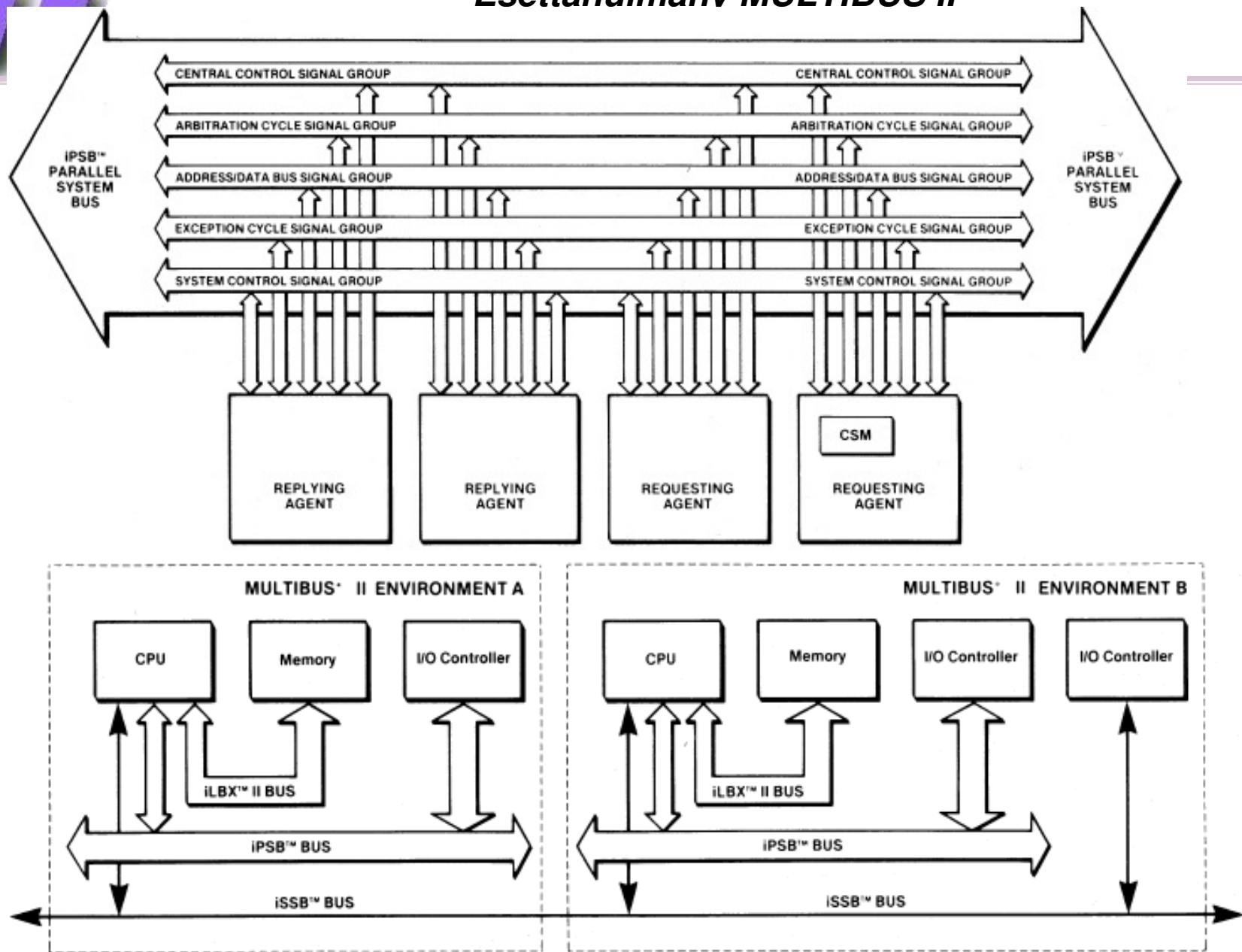
MB-II busz architektúra

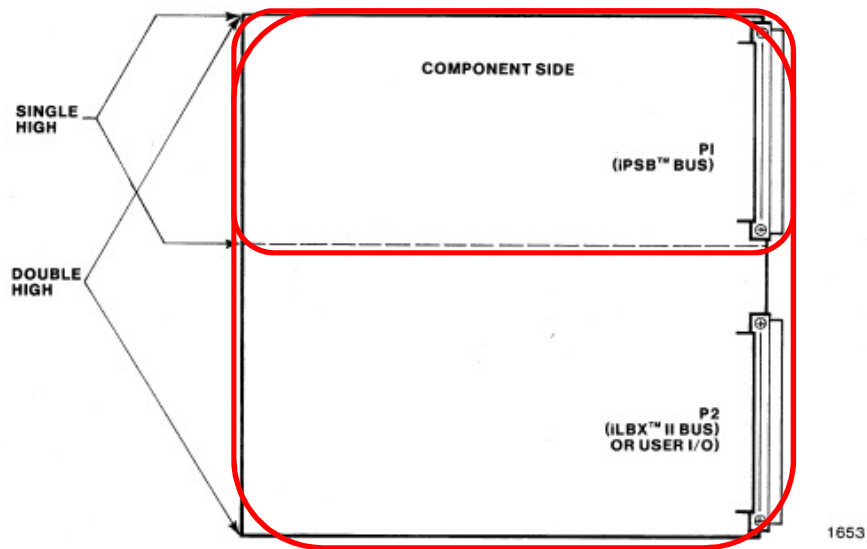
- iPSB párhuzamos rendszerbusz
- iLBX lokális memória kiterjesztés
 - Max. hat kártya
 - Max. két mester
- iSSB soros rendszerbusz
- iSBX lokális I/O kiterjesztés
- iMULTICANELI/O
 - I/O sínkiterjesztés
 - Memória ↔ I/O között





Esettanulmány MULTIBUS II





MB II fizikai méretek

- 6U/3U → 233,3x220mm

- 3U → 100x220mm

→ Eurocard = 100x 160mm

