

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

# FPGA alapú rendszerek fejlesztése

## 9. előadás

Raikovich Tamás

# Tartalom

- **Áramkörön belüli buszrendszerek**
  - Alapfogalmak, topológiák
  - AMBA APB
  - AMBA AXI
- **Megszakítás- és kivételkezelés**

# Buszrendszerek

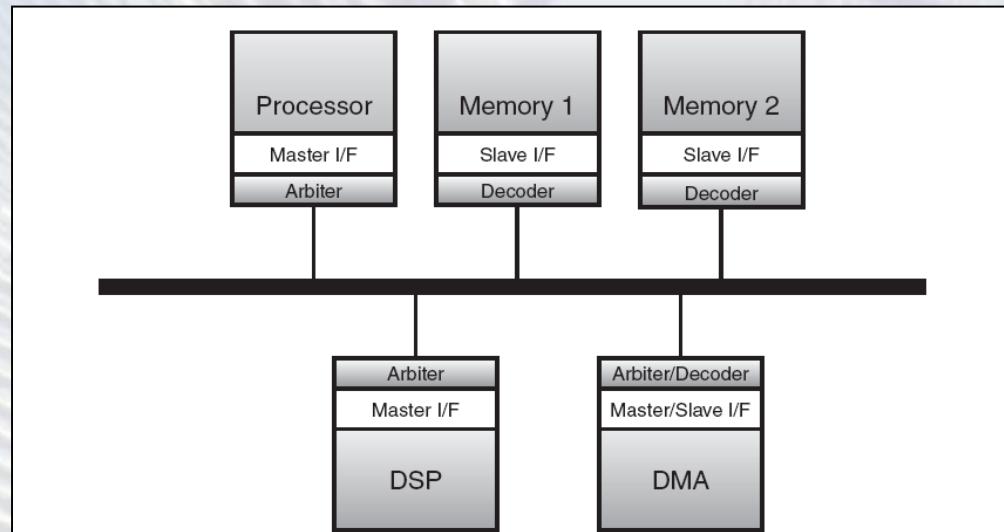
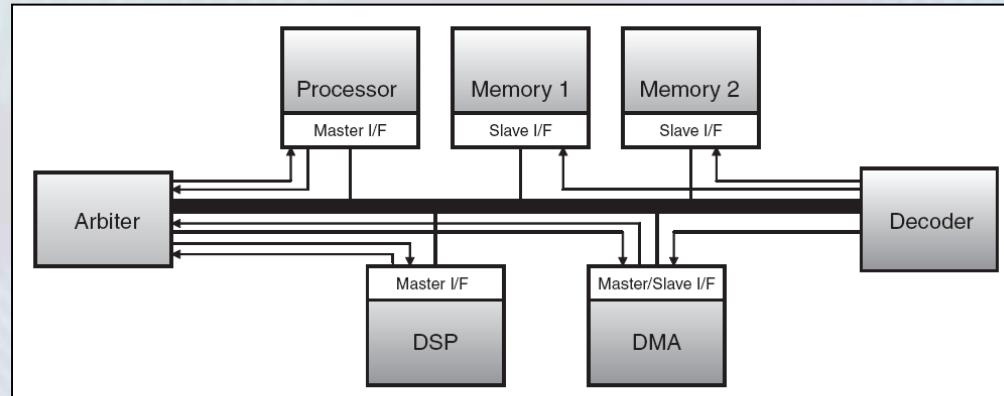
- A busz egy kommunikációs rendszer, amely biztosítja az adatátvitelt az egységek között
- Áramkörön belüli (on-chip) buszok
  - Áramkörön belüli kommunikációhoz ma már nem használunk háromállapotú meghajtókat
    - Egyirányú átvitel → irányonként külön adatvonalak
  - **ARM AMBA:** AHB, APB, AXI
  - IBM CoreConnect: PLB, OPB, DCR
  - Intel (Altera): Avalon
  - OpenCores: Wishbone
- Áramkörön kívüli (off-chip) buszok
  - A következő előadáson lesz ezekről szó

# Buszrendszerek

- **Master (initiator, requester) egység**
  - Buszciklust kezdeményezni tudó egység
- **Slave (target, completer) egység**
  - A buszciklusban csak válaszolni tudó egység
- **Buszciklus (tranzakció) fázisai**
  - Arbitráció: melyik master egység kapja meg a buszvezérlés jogát (arbiter)
  - Cím: slave egység kiválasztása (cím dekóder)
  - Adat: tényleges adatátvitel

# Buszrendszerök

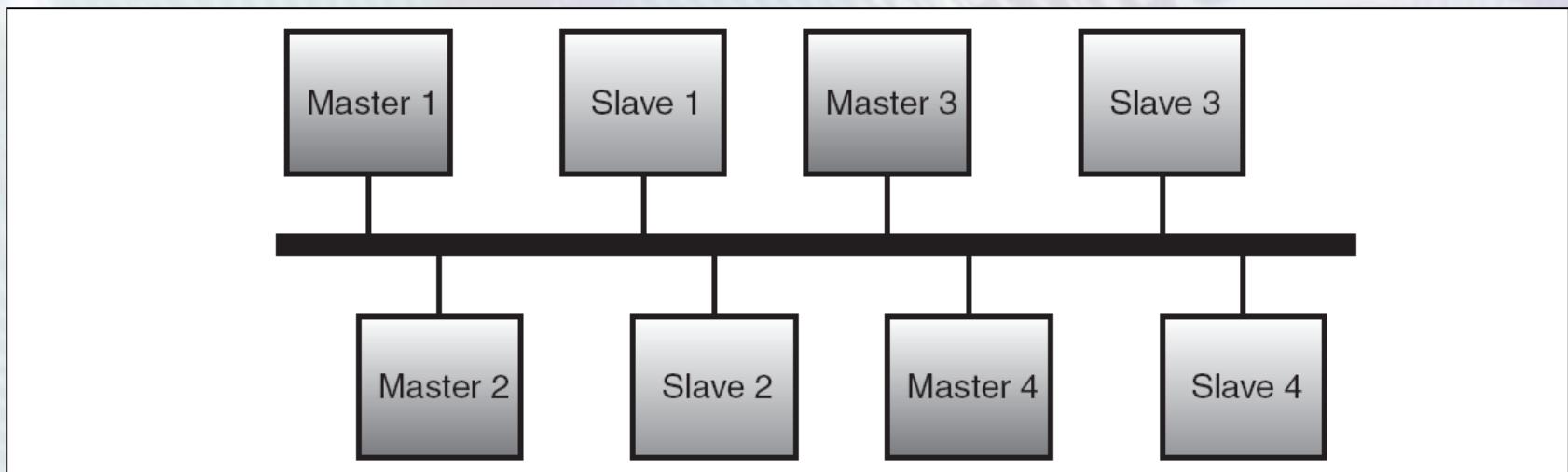
- **Központi arbitráció, dekódolás**
  - Dediált egységek
  - PCI: arbitráció
  - AXI: mindenki
- **Elosztott arbitráció, dekódolás**
  - A master és a slave egységek valósítják meg
  - PCI: dekódolás
  - I<sup>2</sup>C: mindenki



# Buszrendszerek

## Egyszerű busz topológia

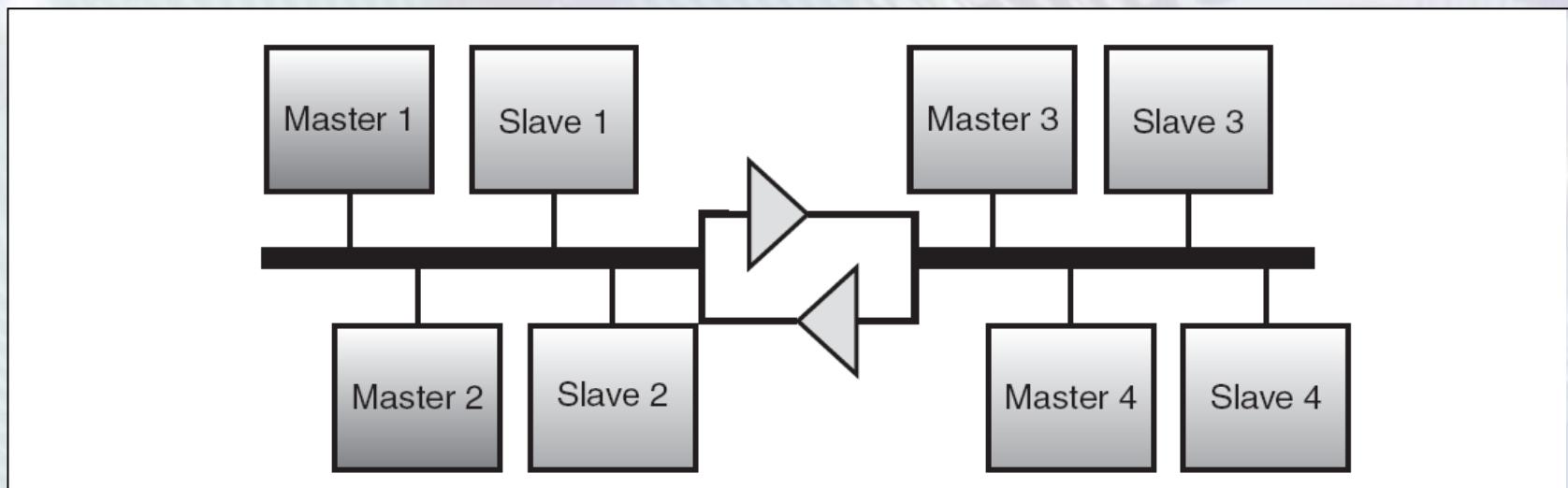
- minden egység egyetlen buszra kapcsolódik
- azonos busz protokoll



# Buszrendszerek

## Megosztott busz topológia

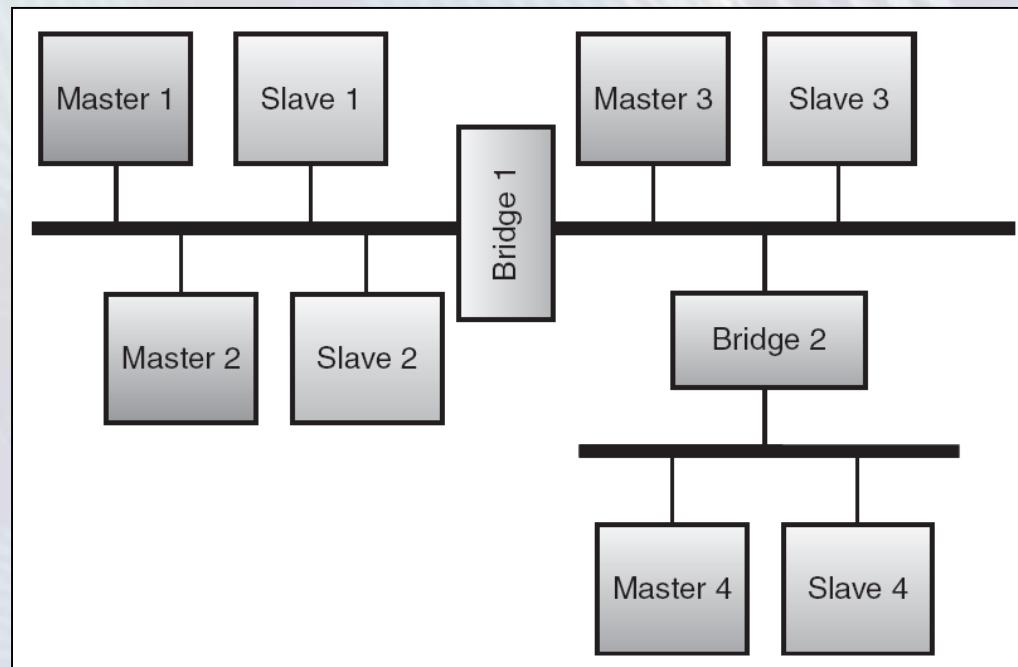
- A részeket jelismétlő áramkör (buffer) választja el
- Azonos busz protokoll



# Buszrendszerök

## Hierarchikus busz topológia

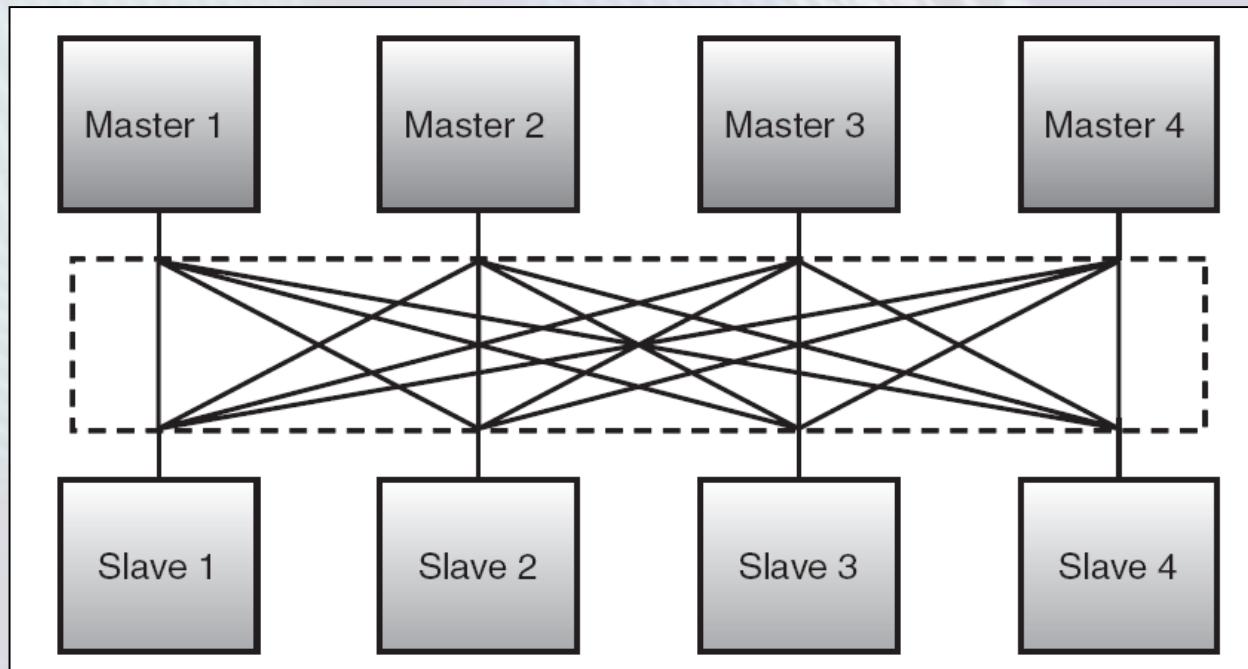
- A részeket híd (bridge) választja el
- Az egyes részek használhatnak eltérő protokollt
- Példa: AHB – APB



# Buszrendszerek

## Pont-pont busz topológia

- Az egységeket crossbar (interconnect, switch) köti össze
- Példa: AXI, PCI-Express



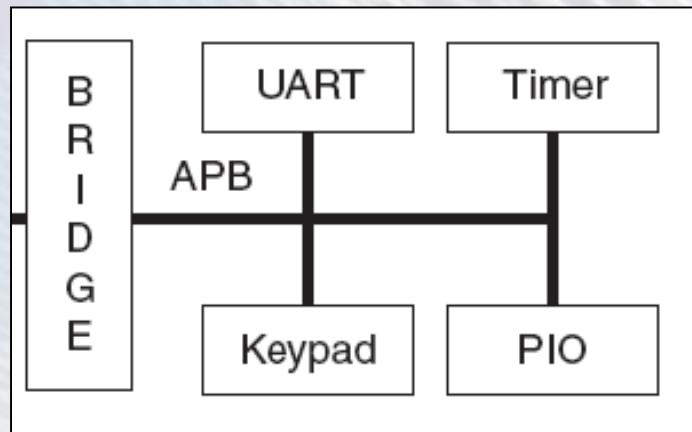
# Advanced Microcontroller Bus Architecture

## Az AXI az ARM AMBA szabvány része

- Advanced Peripheral Bus (APB)
- Advanced High-performance Bus (AHB)
- ***Advanced eXtensible Interface (AXI)***
  - AXI-4, AXI-4 Lite, AXI-4 Stream
  - Részletek: AMBA AXI and ACE Protocol Specification  
AMBA 4 AXI4-Stream Protocol Specification
- AXI Coherency Extensions (ACE)
  - Cache koherencia biztosítása
- Advanced Trace Bus (ATB)

# Advanced Peripheral Bus (APB)

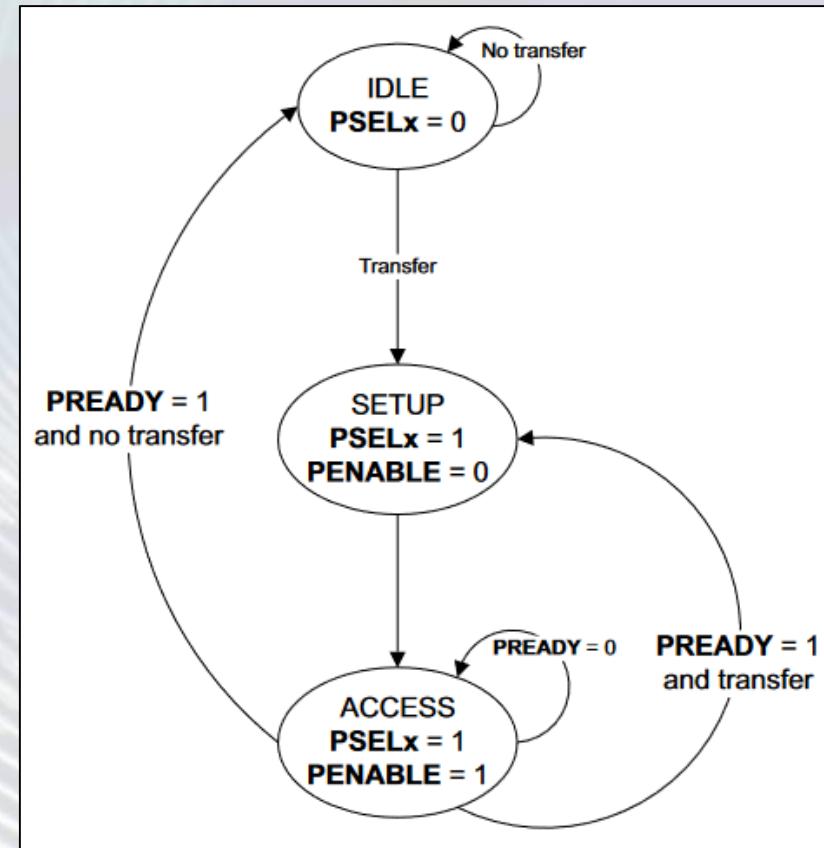
- Egyszerű periféria busz kis sebességű perifériákhoz
- Egy master egység (tipikusan a bridge)
- Több slave egység
- Központi címdekódolás
  - minden slave-hez egyedi kiválasztó jel (PSEL)
- 8, 16 vagy 32 bites adatszélesség
- Egyszavas adatátvitel, nincs burst



# Advanced Peripheral Bus (APB)

## Állapotok

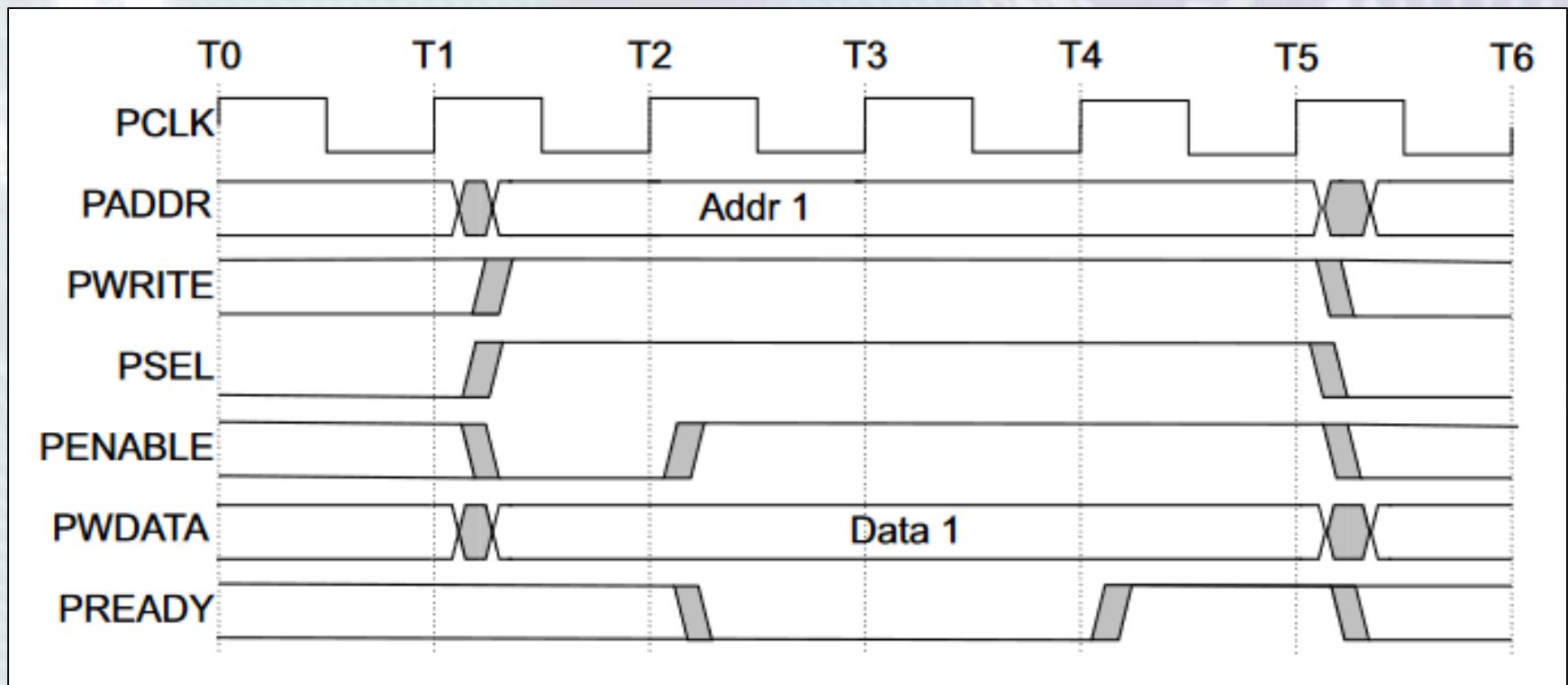
- **IDLE**
  - Nincs adatátvitel
- **SETUP**
  - Cím (PADDR) és irány (PWRITE) érvényes
  - 1 órajelciklus
- **ACCESS**
  - Adatátvitel
  - PREADY-vel nyújtható



# Advanced Peripheral Bus (APB)

## APB írási buszciklus

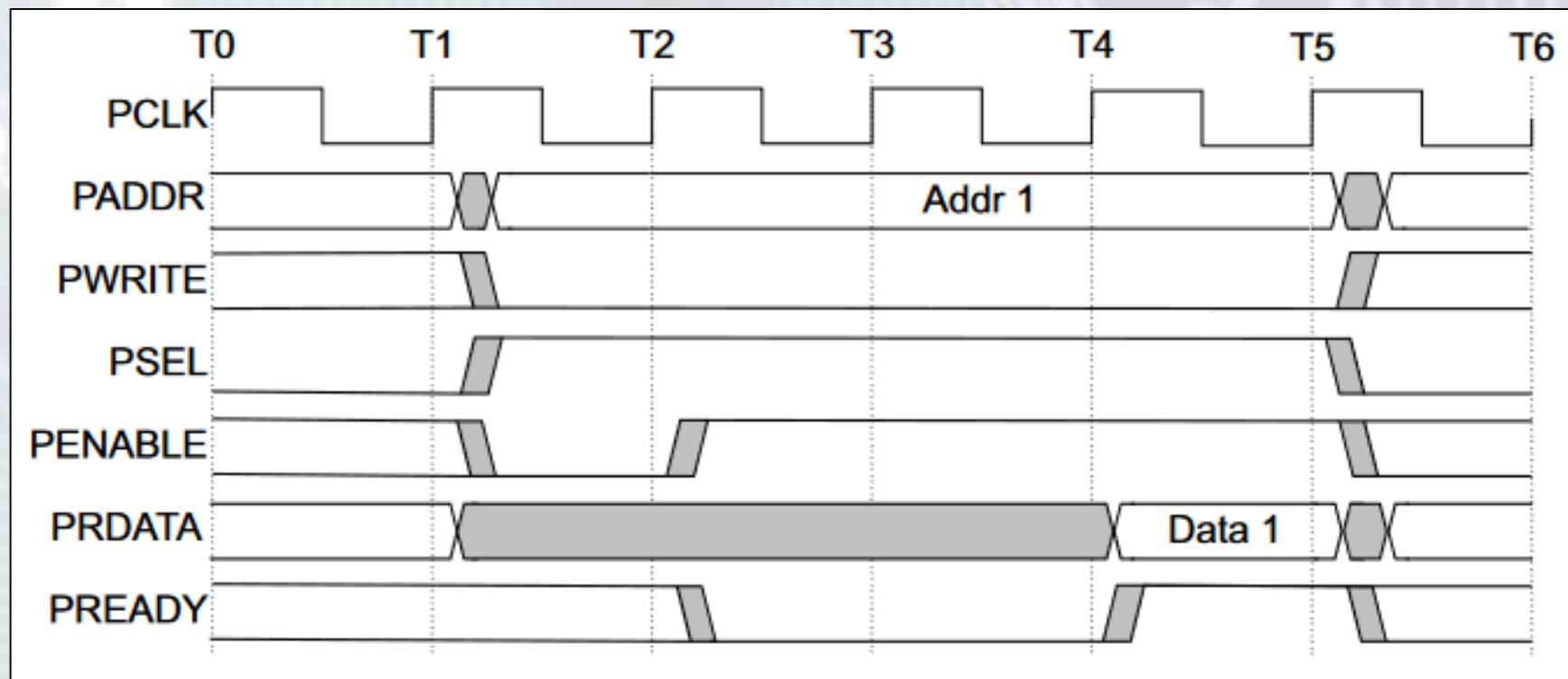
- Az írást a PWRITE magas szintje jelzi
- PWDATA az írási adat



# Advanced Peripheral Bus (APB)

## APB olvasási buszciklus

- Az olvasást a PWRITE alacsony szintje jelzi
- PRDATA az olvasási adat



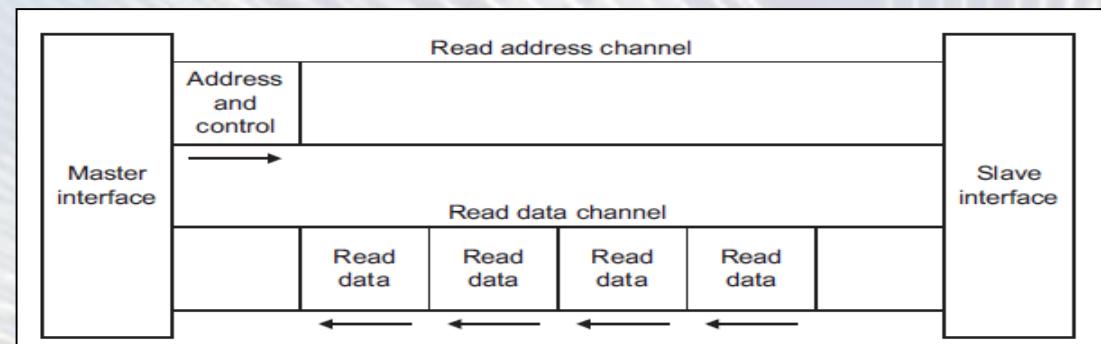
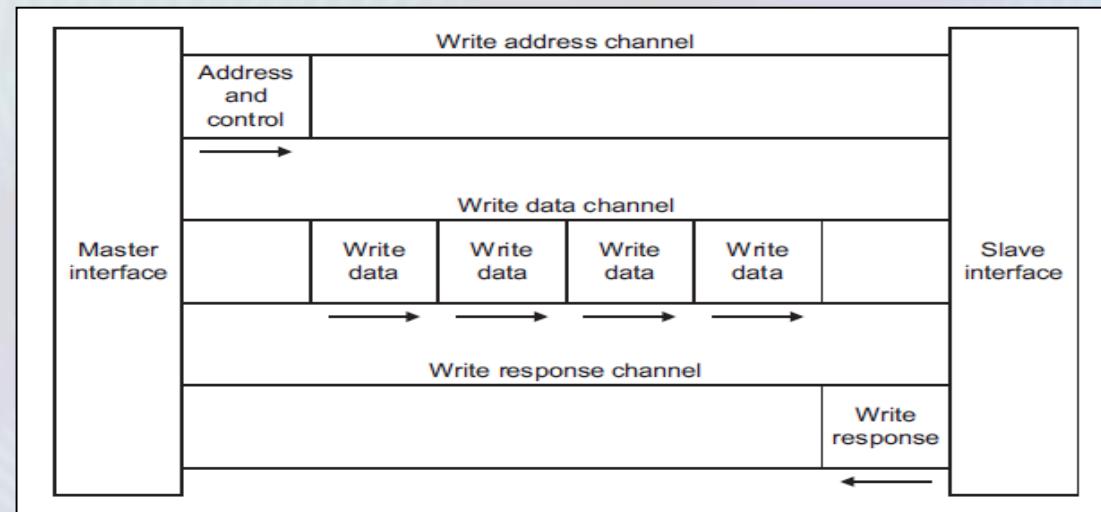
# Advanced eXtensible Interface (AXI)

**Az AXI interfész egységes kapcsolódási felületet nyújt az ARM Cortex és a MicroBlaze processzor alapú beágyazott rendszerek számára**

- **AXI-4**
  - Memóriába leképzett
  - Nagyteljesítményű, burst-ös adatátvitelt biztosít
- **AXI-4 Lite**
  - Memóriába leképzett
  - Egyszerű, nincs burst-ös adatátvitel
- **AXI-4 Stream**
  - Nem memóriába leképzett, FIFO interfész
  - Csak adatok átvitele bust-ös módon

# AXI – Felépítés

- **Öt csatorna**
  - Írási cím
  - Írási adat
  - Írási válasz
  - Olvasási cím
  - Olvasási adat
- **Pont-Pont kapcsolat**
- **A csatornákon egymástól teljesen független időzítés is lehetséges**

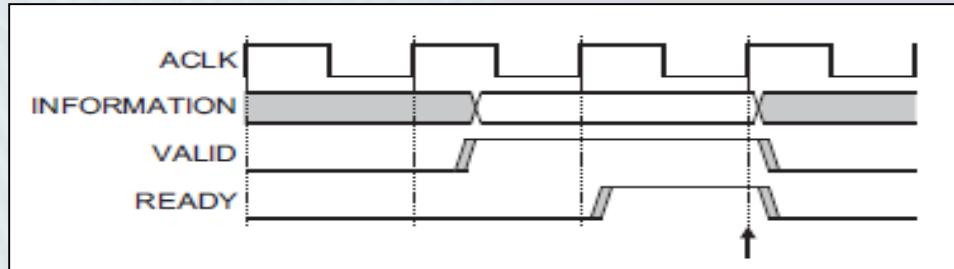


# AXI – Jelek (handshaking)

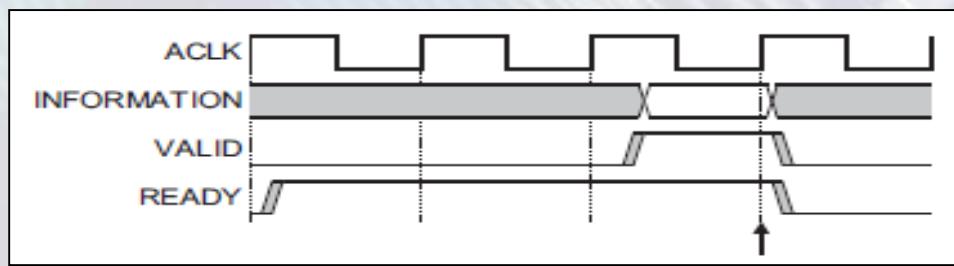
- **Handshake alapú átvitelvezérlés**
  - VALID: a forrás tud adatot küldeni
  - READY: a cél tud adatot fogadni
  - Ha az órajel felfutó élénél minden két aktív, akkor történik meg egy szó átvitele
  - Hasonlóan működik pl. a PCI busz
- **Minden csatornán saját VALID/READY jelpár**
- **Flexibilis funkcionálitás**
  - Várakozási állapotok beszúrása
  - Nyugtázás ugyanabban az órajelciklusban

# AXI – Jelek (handshaking)

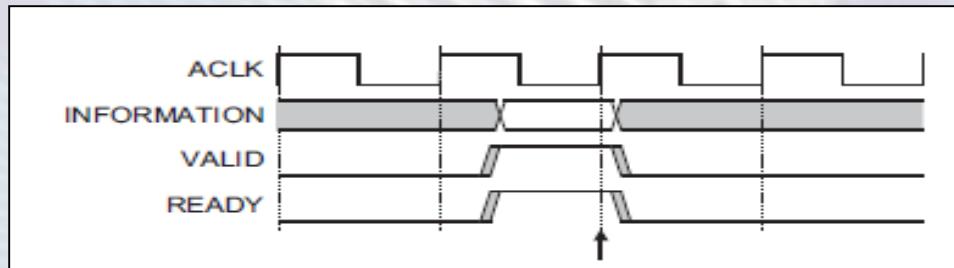
- Cél várakoztat (**VALID** a **READY** előtt)



- Forrás várakoztat (**READY** a **VALID** előtt)



- Nyugtázás ugyanazon órajelciklusban



# AXI – Jelek (handshaking)

A handshake alapú adatátvitel szabályai a holtpont elkerülése végett:

- A forrás nem várhat a READY jelzésre, mielőtt aktiválja a VALID jelzést
- A forrás, ha aktiválta a VALID jelzést, akkor azt nem veheti vissza az aktuális adatátviteli fázis végéig
- A cél várhat a VALID jelzésre, mielőtt aktiválja a READY jelzést
- A cél visszavezeti az aktivált READY jelzést, ha a forrás még nem aktiválta a VALID jelzést

# AXI – Jelek (handshaking)

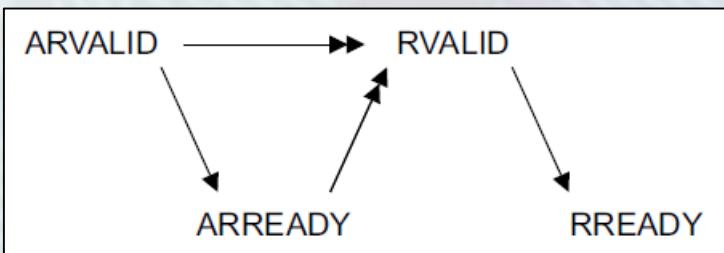
## Függőségek a csatorna handshake jelek között

- **Egyszeres nyíl**

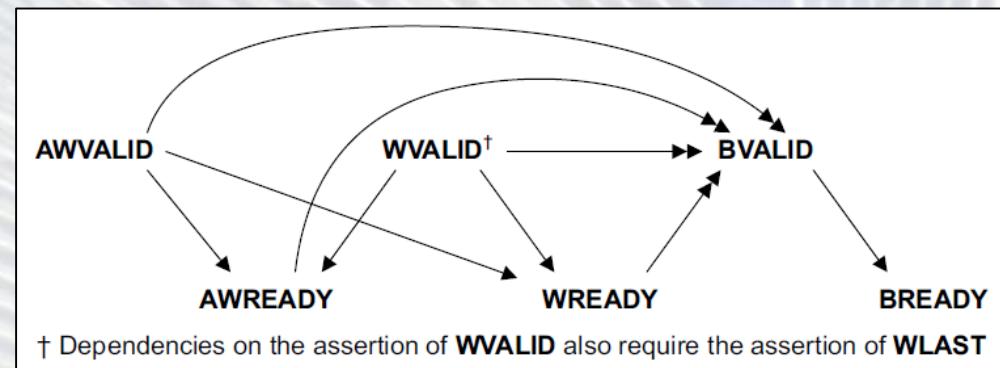
- A cél jel (amerre mutat) a kiindulási jel (amely felől mutat) előtt vagy után is aktiválható
- A forrás jel aktiválása **nem függhet** a cél jel állapotától
- A cél jel aktiválása viszont **függhet** a forrás jel állapotától

- **Dupla nyíl**

- A cél jel csak a kiindulási jel után aktiválható



Olvasási adatátvitel  
handshake függőségek



Írási adatátvitel handshake függőségek

# AXI – Jelek (cím csatornák)

- Az írási és olvasási adatátvitelhez tartozó információk továbbítása
- **ID:** tranzakció azonosító
  - Sorrend felcserélhető (pl. memória vezérlőben)
- **ADDR:** cím információ
- **LEN:** burst méret ( $\text{LEN}[7:0] + 1$  ütem)
  - A burst nem lépheti át a 4KB-os címhatárt
- **SIZE:** bájtszám egy ütemben ( $2^{\text{SIZE}[2:0]}$ )
- **BURST:** a burst típusának jelzése
  - INCR (01): cím növeléses (1 – 256 ütem)
  - WRAP (10): cím átfordulásos (2,4,8,16 ütem)
  - FIXED (00): nincs cím növelés (1 – 16 ütem)

Glob	AXI4	AXI4-Lite
	ACLK	
	ARESETN	
Write Address	AWID	
	AWADDR	
	AWLEN	
	AWSIZE	
	AWBURST	
	AWLOCK	
	AWCACHE	
	AWPROT	
	AWQOS	
	AWREGION	
	AWUSER	
	AWVALID	
	AWREADY	

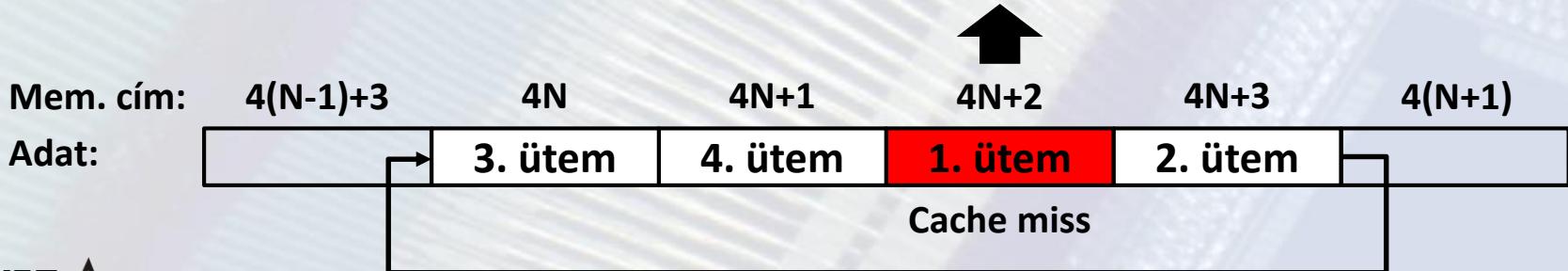
# AXI – Jelek (cím csatornák)

- **LOCK**: kizárolagos, atomi hozzáférés
- **CACHE**: memória típus jelzése
- **PROT**: privilegizált hozzáférés jelzése
- **QOS**: Quality-of-Service támogatás
- **REGION**: több logikai interfész megvalósításának biztosítása a slave oldalon
- **USER**: felhasználó által definiált jelek

	AXI4	AXI4-Lite
ARID		
ARADDR	ARADDR	
ARLEN		
ARSIZE		
ARBURST		
ARLOCK		
ARCACHE	ARCACHE	
ARPROT	ARPROT	
ARQOS		
ARREGION		
ARUSER		
ARVALID	ARVALID	
ARREADY	ARREADY	

# AXI – Jelek (burst típusok)

- **INCR (01)**
  - minden ütemhez egyelőre nagyobb cím tartozik (1 – 256 ütem lehet)
  - normál burst-ös memória elérés esetén
- **FIXED (00)**
  - minden ütemhez ugyanaz a cím tartozik (1 – 16 ütem lehet)
  - például FIFO írás vagy olvasás esetén
- **WRAP (10)**
  - minden ütemhez egyelőre nagyobb cím tartozik, de a határ elérésekor a cím átfordul a tartomány elejére (2, 4, 8 vagy 16 ütem lehet)
  - tipikusan CPU cache vezérlők esetén
    - Cacheline méretű burst-ös adatátvitel (a példában a cacheline méret 4 szó)
    - A kérő (cache-ból hiányzó) adat egyből továbbítható a feldolgozás helyére



# AXI – Jelek (írási adat és válasz csatorna)

- **WDATA**: írási adat (Lite: csak 32 bites)
- **WSTRB**: bájt engedélyező jelek
- **WLAST**: a burst utolsó ütemét jelzi
- **USER**: felhasználó által definiált jelek
- **BID**: a nyugtázott tranzakcióhoz tartozó azonosító (= AWID)
- **BRESP**: nyugta → non-posted írás
  - OKAY (00): adatátvitel OK
  - EXOKAY (01): kizárolagos adatátvitel OK
  - SLVERR (10): slave hiba
  - DECERR (11): címdekódolási hiba

	AXI4	AXI4-Lite
Write Data	WDATA	WDATA(1)
	WSTRB	WSTRB(2)
	WLAST	
	WUSER	
	WVALID	
	WREADY	
Write Response	BID	
	BRESP	BRESP(3)
	BUSER	
	BVALID	
	BREADY	

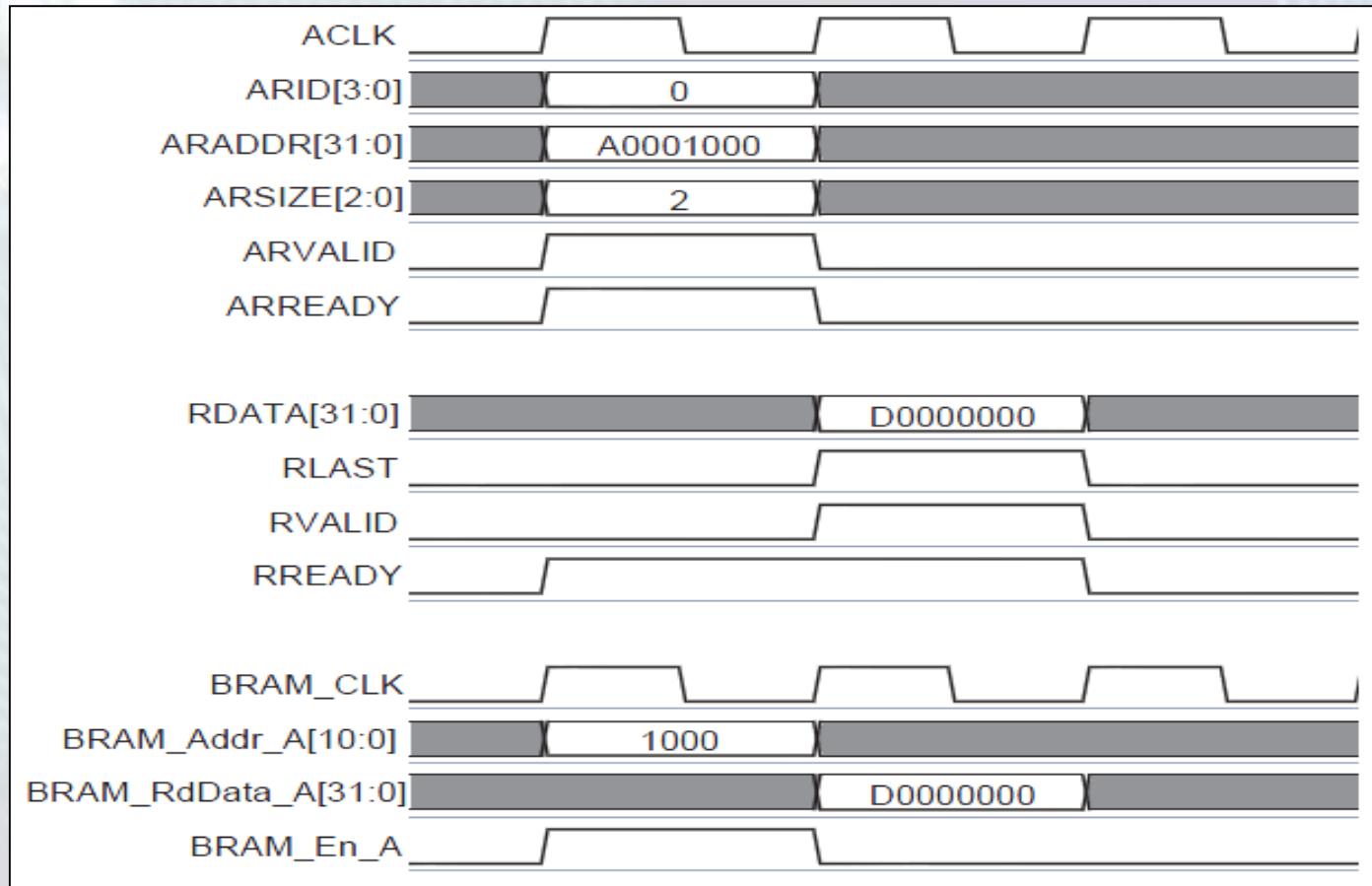
# AXI – Jelek (olvasási adat csatorna)

- **RDATA**: olvasási adat (Lite: 32 bites)
- **RLAST**: a burst utolsó ütemét jelzi
- **RUSER**: felhasználó által definiált jelek
- **RID**: a nyugtázott tranzakcióhoz tartozó azonosító (= ARID)
- **RRESP**: nyugta
  - OKAY (00): adatátvitel OK
  - EXOKAY (01): kizárolagos adatátvitel OK
  - SLVERR (10): slave hiba
  - DECERR (11): címdekódolási hiba

	AXI4	AXI4-Lite
Read Data	RID RDATA RRESP RLAST RUSER	RDATA(1) RRESP(3)
		RVALID
		RREADY

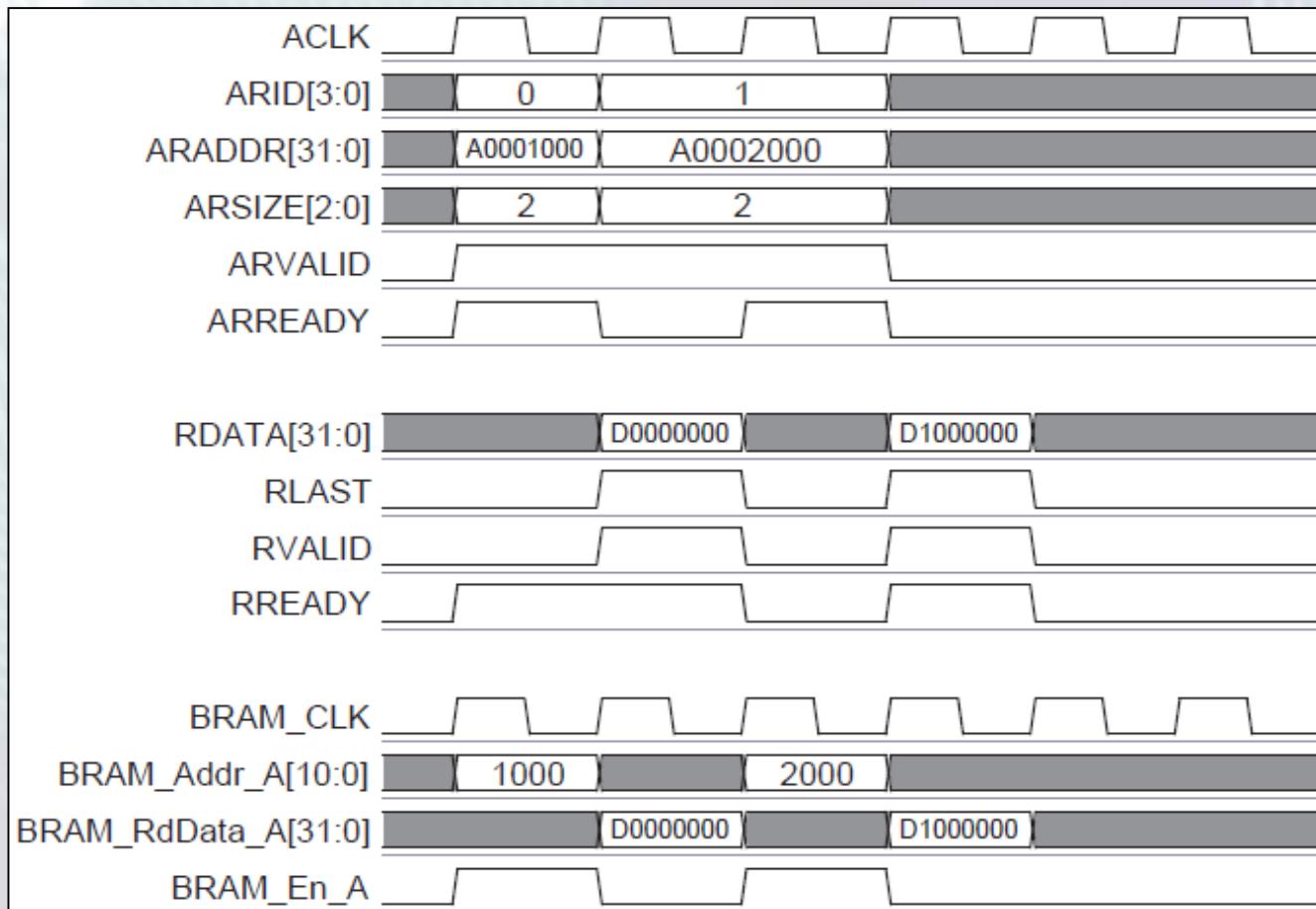
# AXI – Tranzakció példák

## AXI4 BRAM vezérlő: egyszeres olvasás



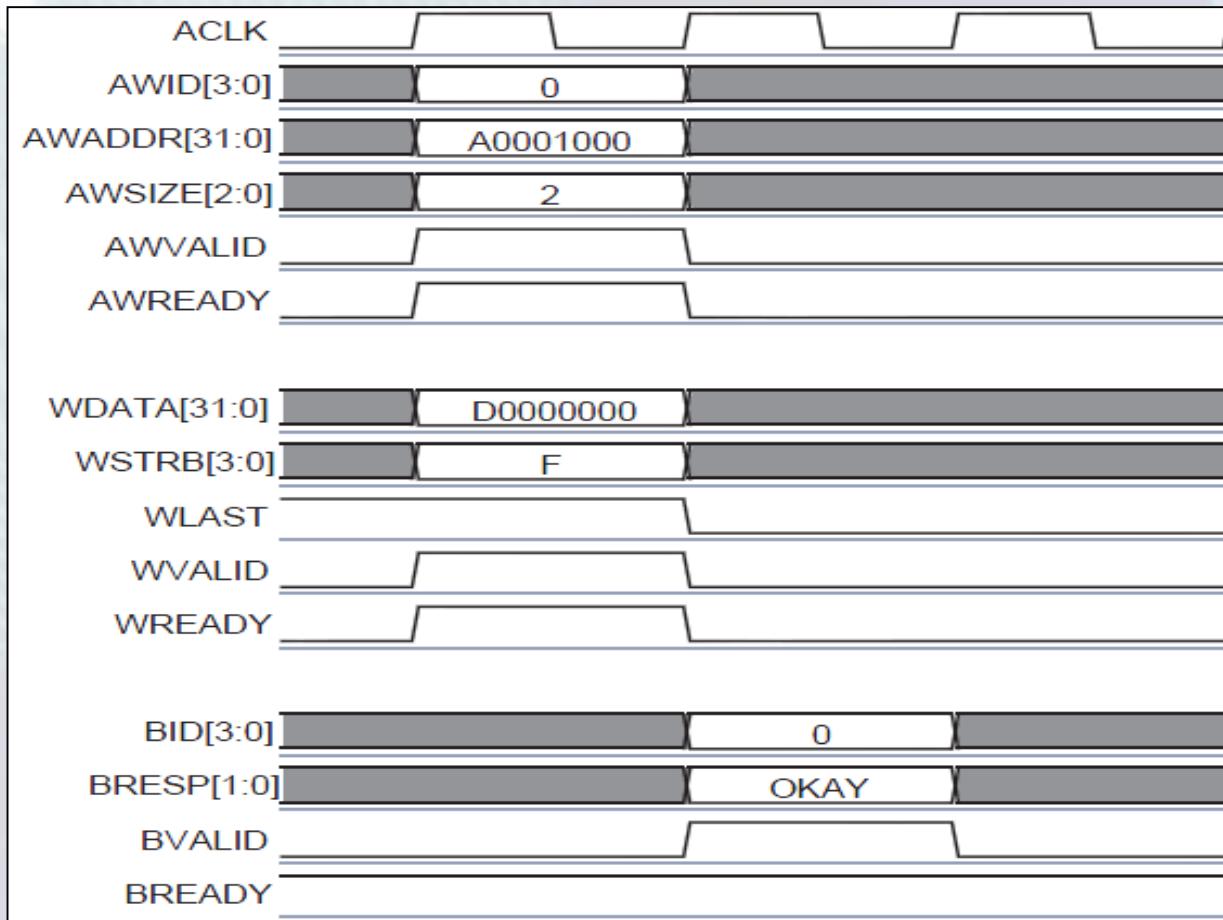
# AXI – Tranzakció példák

## AXI4 BRAM vezérlő: több olvasás egymás után



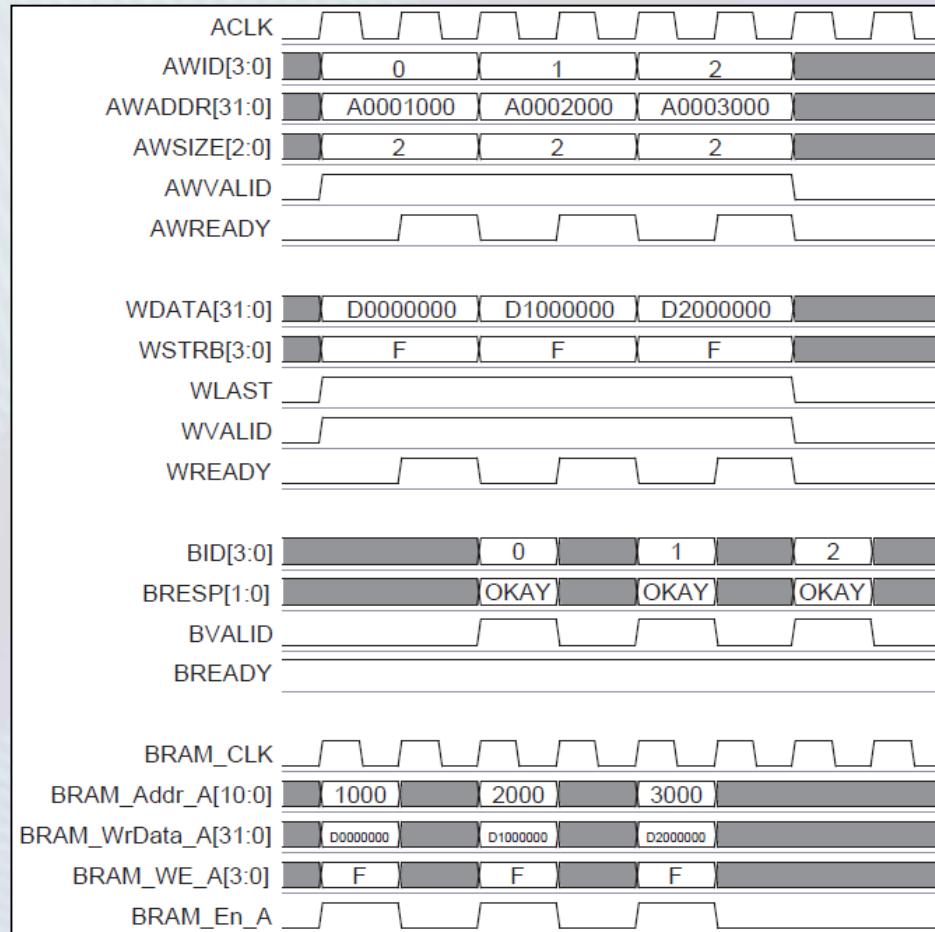
# AXI – Tranzakció példák

## AXI4 BRAM vezérlő: egyszeres írás



# AXI – Tranzakció példák

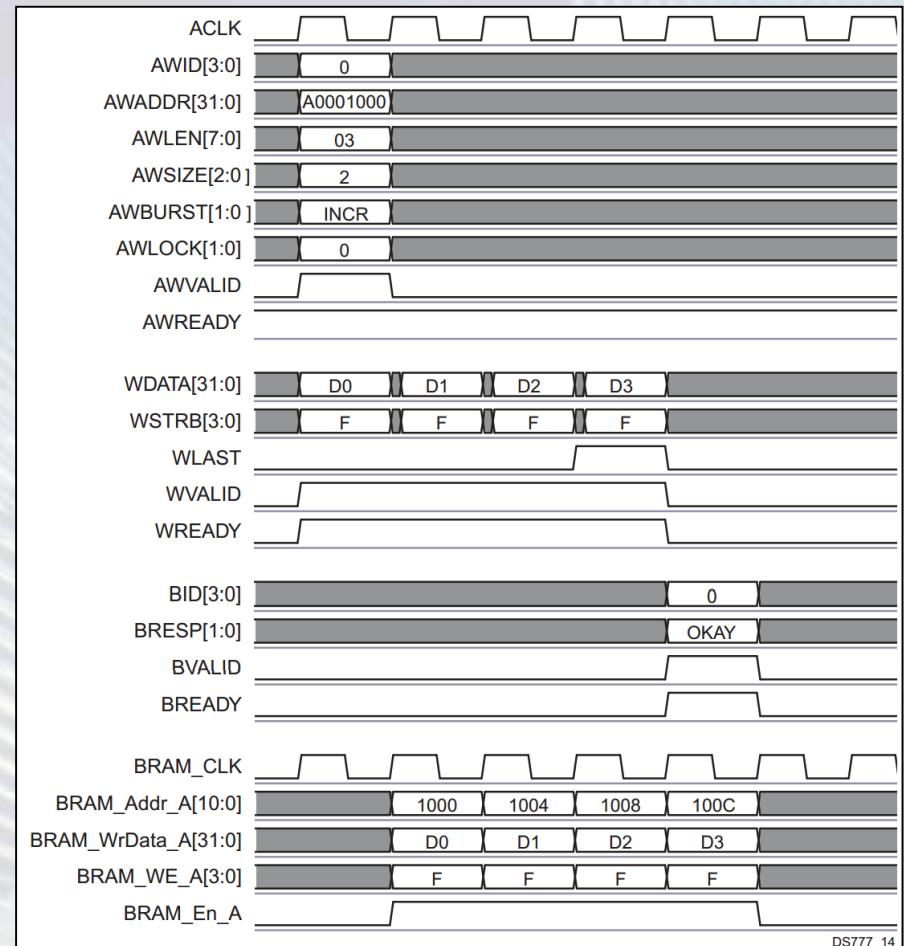
## AXI4 BRAM vezérlő: több írás egymás után



# AXI – Tranzakció példák

## AXI4 BRAM vezérlő: burst-ös írás

- AWSIZE=2 → 32 bites szavak átvitele
- AWLEN=3 → 4 ütem a burst-ös átvitelben
- AWBURST=INCR → a cím minden írás után nő
- A WLAST jel a burst utolsó ütemében aktív



# AXI Interconnect, AXI SmartConnect

- Az AXI Interconnect és az AXI SmartConnect IP-k illesztik egymáshoz az AXI master és az AXI slave perifériákat
  - A slave portokra kapcsolódnak a master IP-k
  - A master portokra kapcsolódnak a slave IP-k
- Az AXI SmartConnect szorosabban integrálódik a Vivado fejlesztői környezetbe
  - Automatikus master és slave IP konfiguráció
  - Adaptálódik a kapcsolódó IP-khez minimális felhasználói beavatkozással
- Részletek
  - AXI Interconnect v2.1 Product Guide (PG059)
  - AXI SmartConnect v1.0 Product Guide (PG247)

# AXI Interconnect, AXI SmartConnect

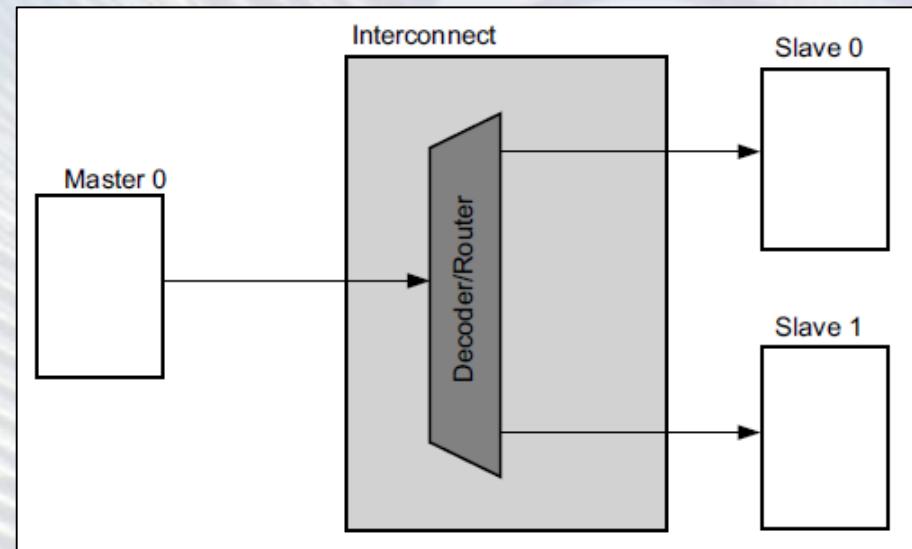
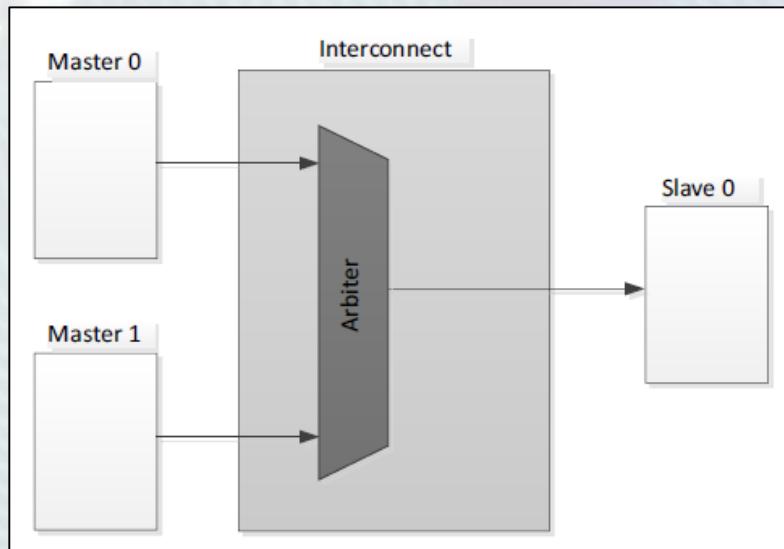
## Szolgáltatások:

- **A slave IP-k címtartományának dekódolása**
  - A slave IP-k felé csak a címtartomány méretének megfelelő számú címbit megy
- **Adatméret konverzió**
- **Eltérő órajel tartományok illesztése**
  - Interconnect: a portokhoz és a crossbar-hoz külön órajel bemenetek tartoznak
  - SmartConnect: magától kitalálja az órajel forrásokat
- **Protokoll konverzió (AXI4, AXI3, AXI4-Lite)**
- **Adat FIFO**
- **Register slice**
  - Pipeline regiszterek a kritikus út csökkentéséhez

# AXI Interconnect, AXI SmartConnect

## Felhasználási esetek

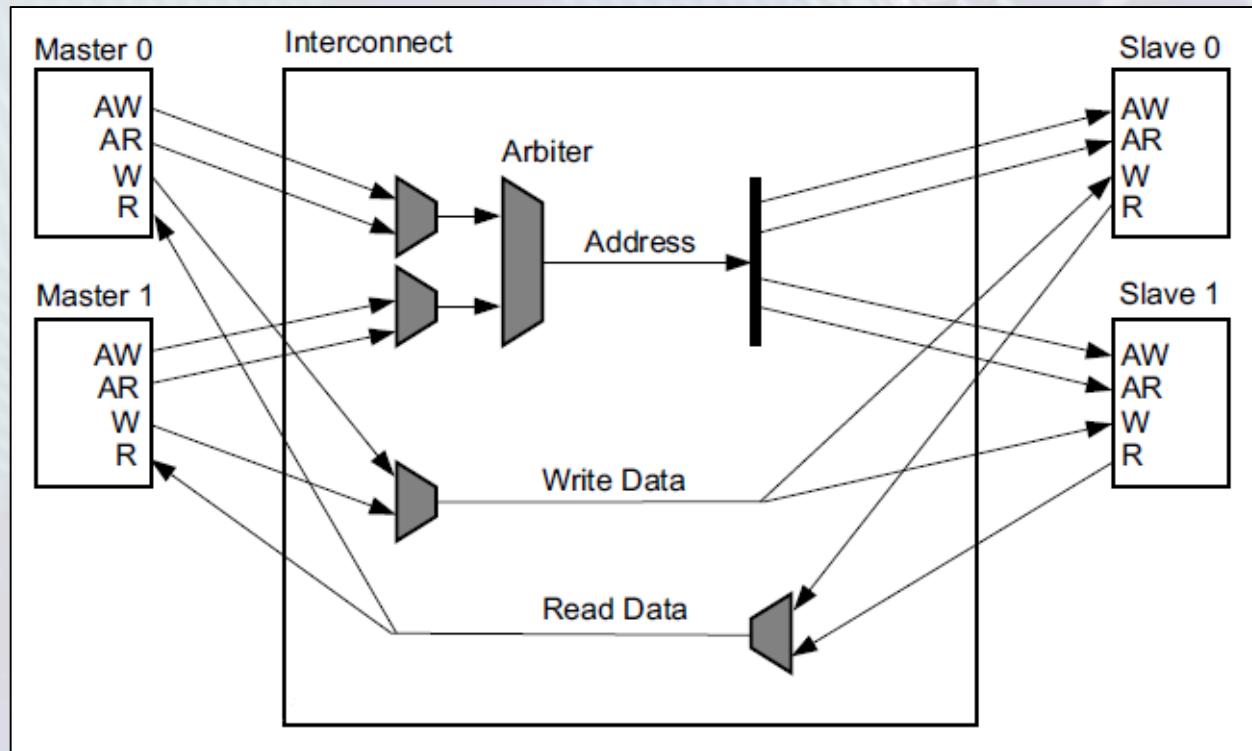
- N master, 1 slave: egyszerű arbiter
- 1 master, N slave: egyszerű dekóder, nincs arbitráció



# AXI Interconnect, AXI SmartConnect

## Felhasználási esetek

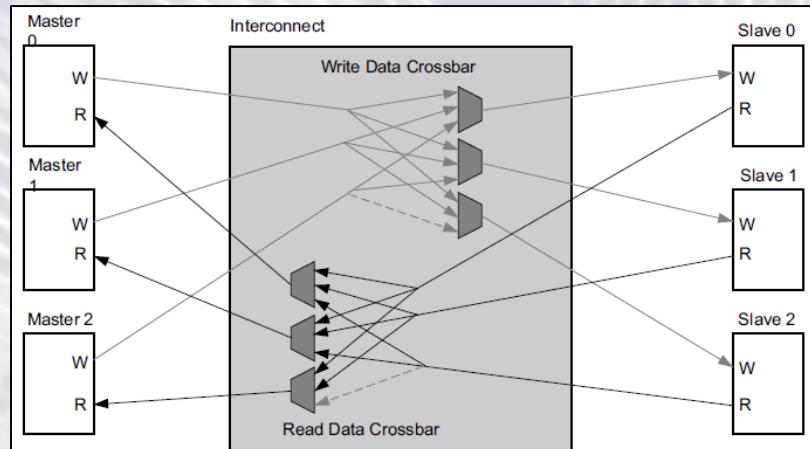
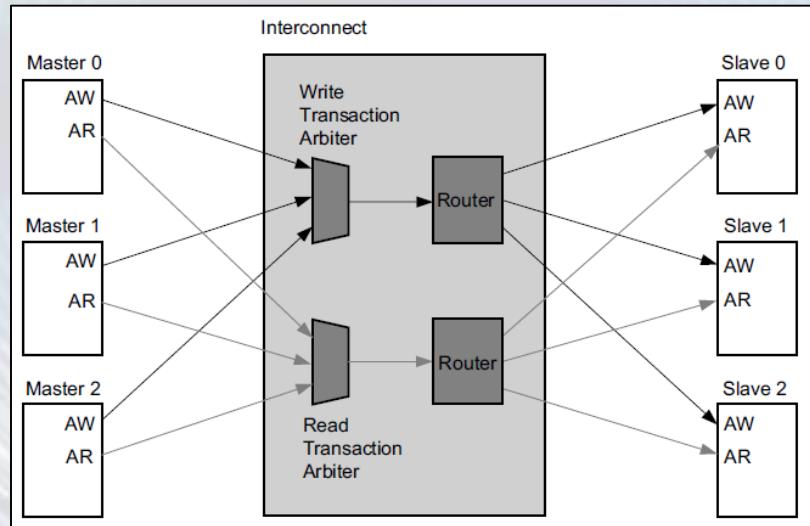
- N master, M slave, osztott elérési mód (egy időben csak egy tranzakció lehet aktív)



# AXI Interconnect, AXI SmartConnect

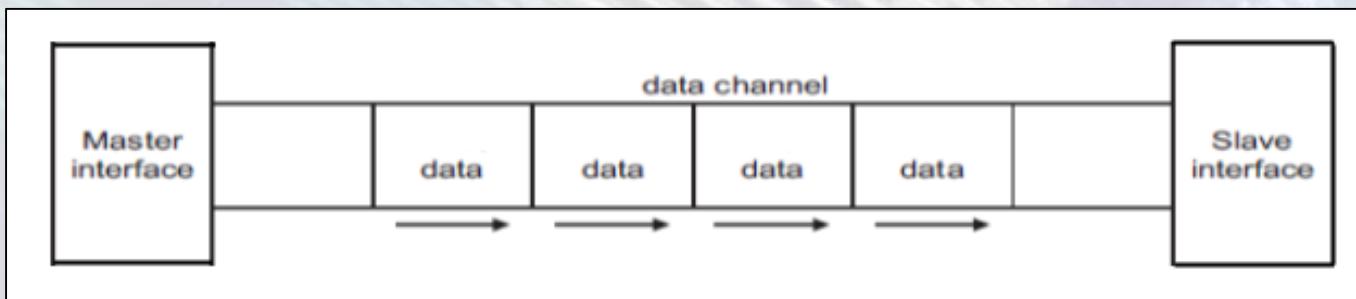
## Felhasználási esetek

- N master, M slave, crossbar
- Egy írási és egy olvasási cím arbiter
- Párhuzamos írási és olvasási adatutak



# AXI Stream – Felépítés

- Nincs cím, egyirányú adatátvitel (master → slave)
  - Az AXI írási adat csatornának feleltethető meg
  - VALID és READY handshake jelek
- A burst méret nem korlátozott
- Folyamok összefésülése (merging), adatméret növelés és csökkentés, null bájtok eltávolítása (packing)
- Adatcsomagok továbbításának támogatása



# AXI Stream – Jelek

- **ACLK**: órajel (szükséges)
- **ARESETn**: aktív alacsony aszinkron reset jel (szükséges)
- **TVALID**: a forrás (master) tud adatot küldeni (szükséges)
- **TREADY**: a cél (slave) tud adatot fogadni
- **TDATA[ $(8N-1):0$ ]**: a továbbított adat
- **TSTRB[ $(N-1):0$ ]**: adat vagy pozíció bájt jelzése
- **TKEEP[ $(N-1):0$ ]**: jelzi, hogy a bájt része-e a folyamnak
- **TLAST**: az adatcsomag végének jelzése
- **TID**: adatfolyam azonosító
- **TDEST**: cél azonosító (útvonal választáshoz)
- **TUSER**: felhasználó által definiált jelek

# AXI Stream – Jelek (bájt típusok)

- **TKEEP[i]=1, TSTRB[i]=1: adatbájt**
  - Az adott bájt érvényes információt tartalmaz
  - Továbbítani kell a forrástól a cél felé
- **TKEEP[i]=1, TSTRB[i]=0: pozíció bájt**
  - Az adott bájt az adatbájtok relatív pozícióját határozza meg az adatfolyamban
  - Továbbítani kell a forrástól a cél felé, de mivel csak helykitöltő, így az értéke megváltoztatható
- **TKEEP[i]=0, TSTRB[i]=0: null bájt**
  - Az adott bájt semmilyen információt sem hordoz
  - Eltávolítható a folyamból, beszúrható a folyamba
- **TKEEP[i]=0, TSTRB[i]=1: tiltott kombináció**

# Xilinx AXI Stream IP-k

- Részletek: **AXI4-Stream Infrastructure IP Suite (PG085)**
- **AXI4-Stream Broadcaster**
  - A bemeneti master folyamot megismétli több (2 – 16) kimeneti slave folyamon
- **AXI4-Stream Combiner**
  - Több (2 – 16) bemeneti adatfolyam egyesítése egy szélesebb kimeneti adatfolyamban
- **AXI4-Stream Clock Converter**
  - Eltérő órajel tartományok illesztését biztosítja
- **AXI4-Stream Data FIFO**
- **AXI4-Stream Data Width Converter**
  - TDATA méret növelés, csökkentés (1:N, N:1, M:N)

# Xilinx AXI Stream IP-k

- **AXI4-Stream Register Slice**
  - Pipeline regiszterek a kritikus úthossz csökkentéséhez
- **AXI4-Stream Subset Converter**
  - Eltérő opcionális jelkészlettel rendelkező AXI Stream interfések illesztését biztosítja
- **AXI4-Stream Switch**
  - Útválasztást biztosít master és slave egységek között
- **AXI4-Stream Interconnect**
  - A Broadcaster és a Combiner kivételével a fenti IP-ket foglalja egyetlen konfigurálható egységbe

# Tartalom

- Áramkörön belüli buszrendszerek
  - Alapfogalmak, topológiák
  - AMBA APB
  - AMBA AXI
- *Megszakítás- és kivételezelés*

# Megszakítás, kivétel

- Az utasítások végrehajtása (alapvetően) a programozó által meghatározott sorrendben történik
  - Események kezelése lekérdezéssel → lassú
  - Sok esetben gyorsabb reagálás kell → megszakítás
- Megszakítás (interrupt)
  - Jelzés a processzornak kiszolgálási igényre
  - A CPU az aktuális utasítás végrehajtása után elfogadhatja
  - Hardver megszakítás
    - Jelzés külső hardver egységtől
  - Szoftver megszakítás
    - Kivételes esemény a CPU-n belül (kivétel, exception)
      - A processzor belső hibás állapotának jelzése
    - Speciális, megszakítást okozó utasítás (pl. x86 → INT x)

# MicroBlaze V processzor – Események

- **Vektortáblázat a 0x00000000 címtől, áthelyezhető**

Type	Vector Address	Description
Reset	C_BASE_VECTORS + 0x0	Execution starts from here after reset is released.
Trap	C_BASE_VECTORS + 0x4	Default address for exceptions and interrupts

- **Események és prioritási sorrendjük (a RISC-V specifikáció szerint)**
  1. Reset (MicroBlaze V: Reset és Debug\_Rst portok)
  2. Nem maszkolható megszakítás (MicroBlaze V: Ext\_NM\_BRK port)
  3. Platform megszakítás (MicroBlaze V: Ext\_BRK port)
  4. Külső megszakítás (MicroBlaze V: Interrupt port)
  5. Utasítás töréspont kivétel
  6. Utasítás laphiba kivétel
  7. Utasítás elérési hiba kivétel
  8. Illegális utasítás, nem igazított utasítás cím, rendszerhívás (environment call) és adat töréspont kivételek
  9. Nem igazított adatmemória olvasás és írás cím kivételek
  10. Adatmemória olvasás és írás laphiba kivételek
  11. Adatmemória olvasás és írás elérési hiba, AXI4-Stream kivételek

# MicroBlaze V processzor – Események

- Néhány fontosabb RISC-V vezérlő/státusz regiszter (CSR)
  - *mstatus*: mód és státusz bitek, globális megszakítás eng. (MIE)
  - *mtvec*: vektortáblázat báziscím (C\_BASE\_VECTORS kezdőérték)
  - *mepc*: a megszakított utasítás címe
  - *mcause*: az aktuális megszakítás/kivétel oka
  - *mie, mip*: megszakítás engedélyező és flag regiszterek
- Az események direkt kezelése
  - Ugrás az *mtvec* regiszterben tárolt címre
  - A kiszolgáló rutinnak kell azonosítania és kezelnie az egyes eseményeket
  - Egyszerűen megvalósítható, de nem hatékony
- Az események vektoros kezelése
  - Az *mtvec* regiszter a vektortáblázatra mutat
  - minden forráshoz tartozik egy bejegyzés (eseménykezelő)
  - Hatékonyabb megoldás

# MicroBlaze V processzor – Megszakítás

- Az **Interrupt** bemenet lehet él- vagy szintérzékeny
  - A **C\_INTERRUPT\_IS\_EDGE** paraméterrel állítható be
  - Szintérzékeny megszakítás bemenetet
    - Aktívan kell tartani az esemény elfogadásáig
    - A forrásnál nyugtázni kell az eseményt
  - Élérzékeny megszakítás bemenet
    - A processzor tárolja az eseményt **MIE** értékétől függetlenül
    - Érvényre jut, ha **MIE=1**
- Normál mód
  - Egyetlen megszakítás vektor van
  - Szoftveres forrás azonosítás és nyugtázás szükséges
- Alacsony késleltetésű (gyors) mód
  - A forrástól függő vektort a megszakítás vezérlő adja
  - A processzor nyugtázó jelet küld a megszakítás vezérlőnek

# AXI Interrupt Controller

- **Megszakítás vezérlő periféria**
- **Biztosítja max. 32 hardver megszakítás forrás kezelését**
  - A 0. megszakítás bemenet a legnagyobb prioritású
  - Szinkronizálja a megszakítás bemeneteket a saját órajeléhez
  - Él- vagy szintérzékeny megszakítás bemenetek (a kapcsolódó IP *irq* megszakítás interfészének *SENSITIVITY* paramétere)
    - Felfutó élre: *SENSITIVITY = EDGE\_RISING*
    - Lefutó élre: *SENSITIVITY = EDGE\_FALLING*
    - *Magas szintre:* *SENSITIVITY = LEVEL\_HIGH*
    - Alacsony szintre: *SENSITIVITY = LEVEL\_LOW*
- **A megszakításkérő kimenet lehet él- vagy szintérzékeny**
- **Biztosít szoftver megszakításokat is**
- **Támogatja a MicroBlaze V CPU gyors megszakítás módját**
- **Több megszakítás vezérlő kaszkádosítható**

# AXI Interrupt Controller

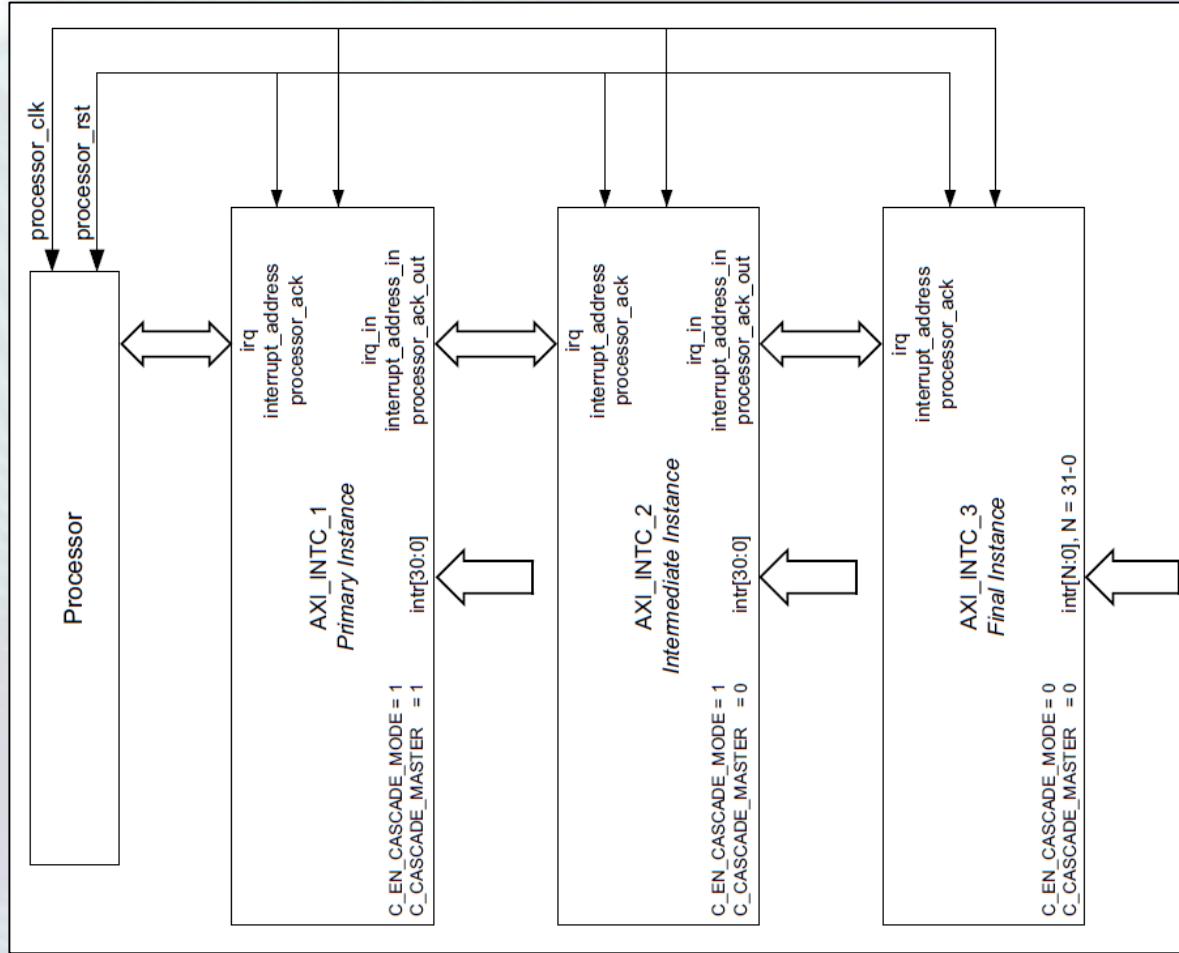
- Részletek: AXI Interrupt Controller Product Guide (PG099)
- A megszakítás vezérlő regiszterkészlete

Address Offset	Register Name	Description
00h	ISR	Interrupt Status Register (ISR)
04h	IPR	Interrupt Pending Register (IPR)
08h	IER	Interrupt Enable Register (IER)
0Ch	IAR	Interrupt Acknowledge Register (IAR)
10h	SIE	Set Interrupt Enables (SIE)
14h	CIE	Clear Interrupt Enables (CIE)
18h	IVR	Interrupt Vector Register (IVR)
1Ch	MER	Master Enable Register (MER)
20h	IMR	Interrupt Mode Register (IMR)
24h	ILR	Interrupt Level Register (ILR)
100h to 170h	IVAR	Interrupt Vector Address Register (IVAR)

- Az ide tartozó szoftveres rész a gyakorlaton lesz

# AXI Interrupt Controller

## Megszakítás vezérlők kaszkádosítása



# Megszakítások kezelése (SW)

- Lépések
  - A megszakításkezelő rutin regisztrálása
    - A perifériához tartozik egy megszakítás azonosító érték
  - A megszakítás vezérlő konfigurálása
    - Globális megszakítás engedélyezés
    - A megfelelő megszakítás bemenet engedélyezése
  - Megszakítás engedélyezése a MicroBlaze V processzoron
  - A megszakításos módon kezelt perifériák
    - Konfigurálása
    - A megszakítások engedélyezése
- A megszakításkezelő rutinban használt globális változók
  - *volatile* módosító: optimalizálás letiltása a változóra
  - Nem atomi műveletek (pl. `i++`): megszakítások tiltása, művelet végrehajtása, megszakítások engedélyezése (szinkronizáció)