



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

Mikrorendszerek tervezése

AXI interfész

Fehér Béla
Raikovich Tamás



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)

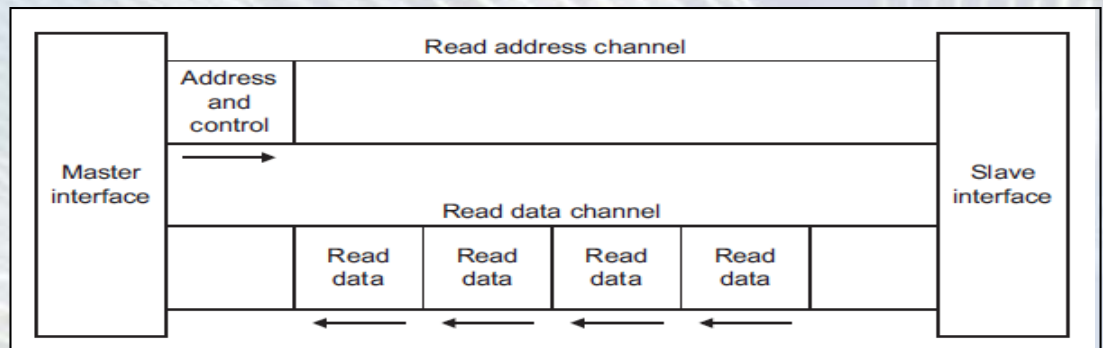
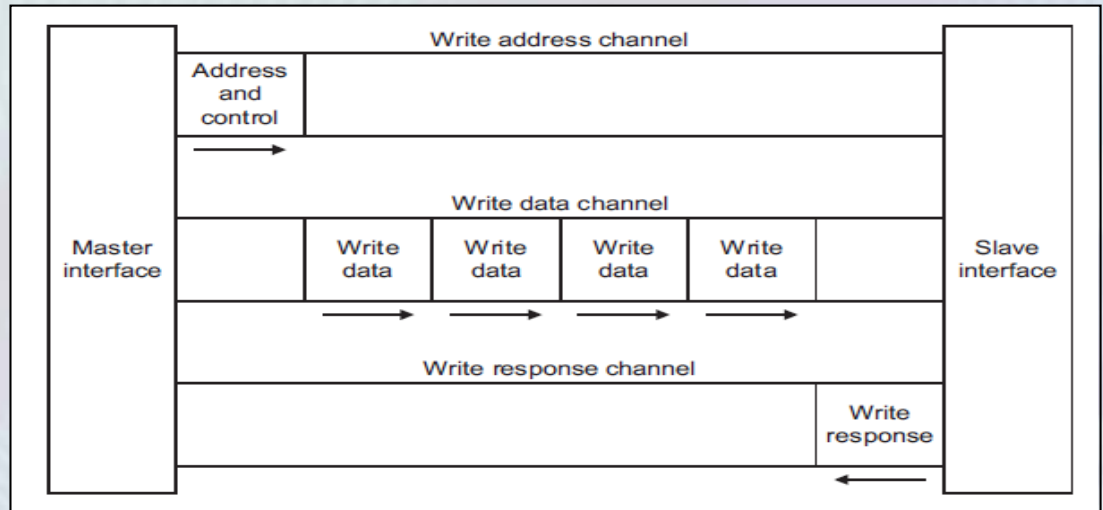
Advanced eXtensible Interface (AXI)

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 burst-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

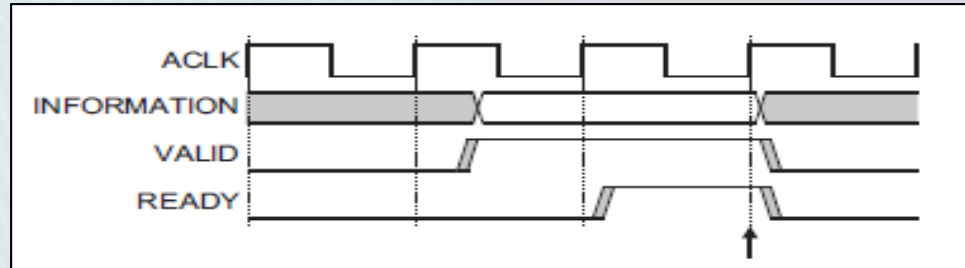


AXI – Jelek (handshaking)

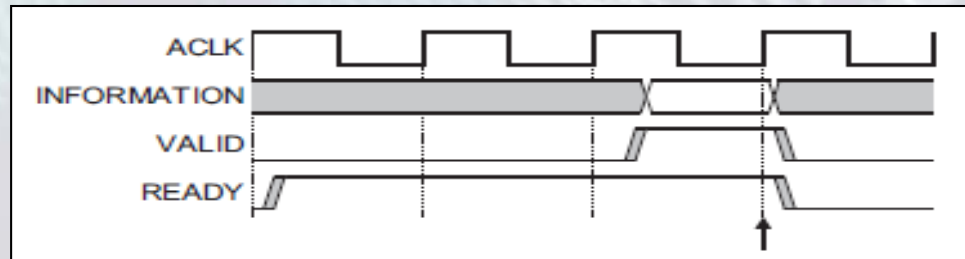
- **Handshake alapú átvitelvezérlés**
 - VALID: a forrás tud adatot küldeni
 - READY: a cél tud adatot fogadni
 - Ha az órajel felfutó élénél mindkettő aktív, akkor történik meg egy szó átvitele
 - Hasonlóan működik pl. a PCI busz
- **Minden csatornán saját VALID/READY jelpár**
- **Flexibilis funkcionalitás**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

AXI – Jelek (handshaking)

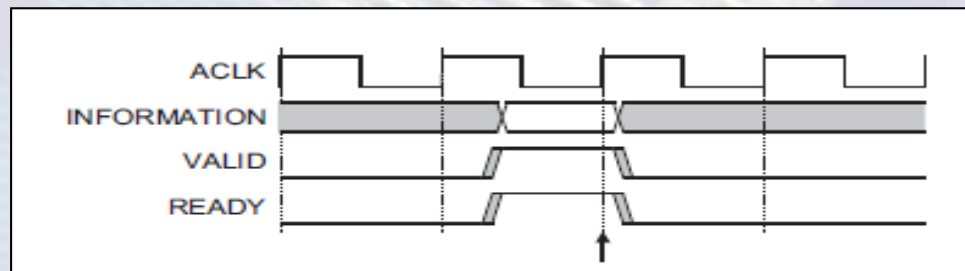
- Cél várakoztat (VALID a READY előtt)



- Forrás várakoztat (READY a VALID előtt)



- Nyugtázás ugyanazon órajelciklusban



AXI – Jelek (handshaking)

A handshake alapú adatátvitel szabályai a holtpont elkerülése végett:

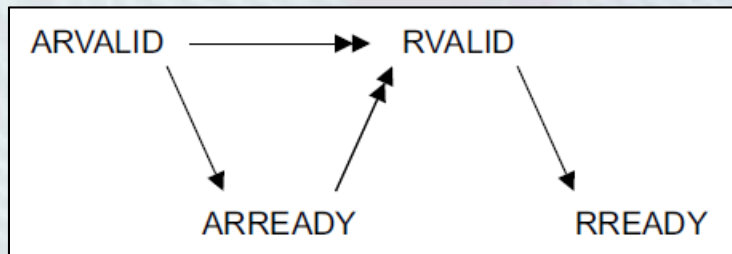
- **A forrás nem várhat a READY jelzésre, mielőtt aktiválja a VALID jelzést**
- **A forrás, ha aktiválta a VALID jelzést, akkor azt nem veheti vissza az aktuális adatátviteli fázis végéig**
- **A cél várhat a VALID jelzésre, mielőtt aktiválja a READY jelzést**
- **A cél visszaveheti az aktivált READY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



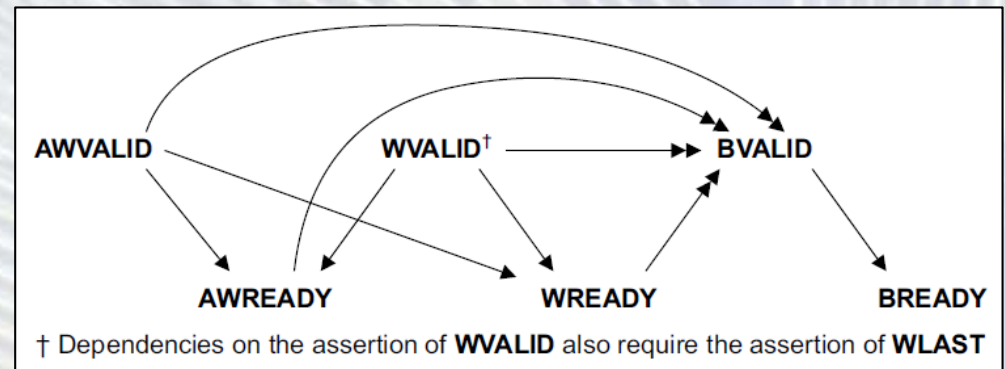
AXI – Jelek (handshaking)

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 ($LEN[7:0] + 1$ ütem)
 - A burst nem lépheti át a 4KB-os címhatárt
- **SIZE**: bájtszám egy ütemben ($2^{SIZE[2:0]}$)
- **BURST**: a burst típusának jelzése
 - INCR (01): cím növeléses (1 – 256 ütem)
 - WRAP (10): cím átfordulós (2,4,8,16 ütem)
 - FIXED (00): nincs cím növelés (1 – 16 ütem)

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

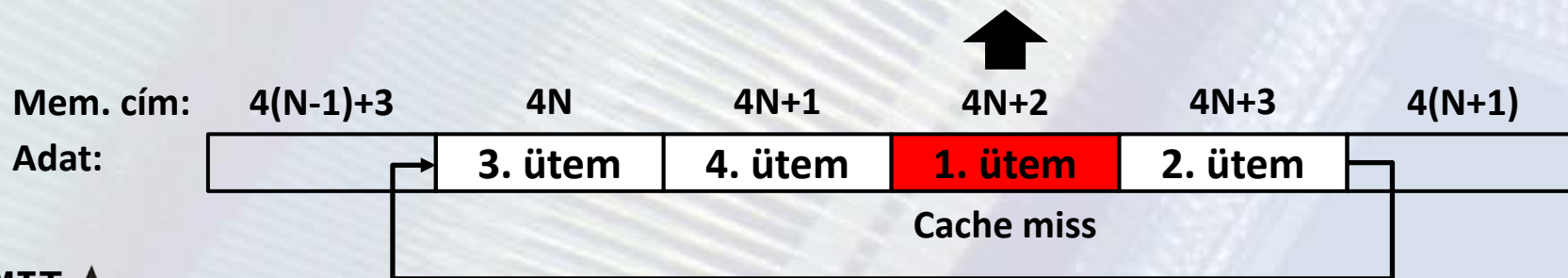
AXI – Jelek (cím csatornák)

- **LOCK**: kizárólagos, 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

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

AXI – Jelek (burst típusok)

- **INCR (01)**
 - Minden ütemhez eggyel 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 eggyel 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 kért (cache-ből 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): kizárólagos 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	

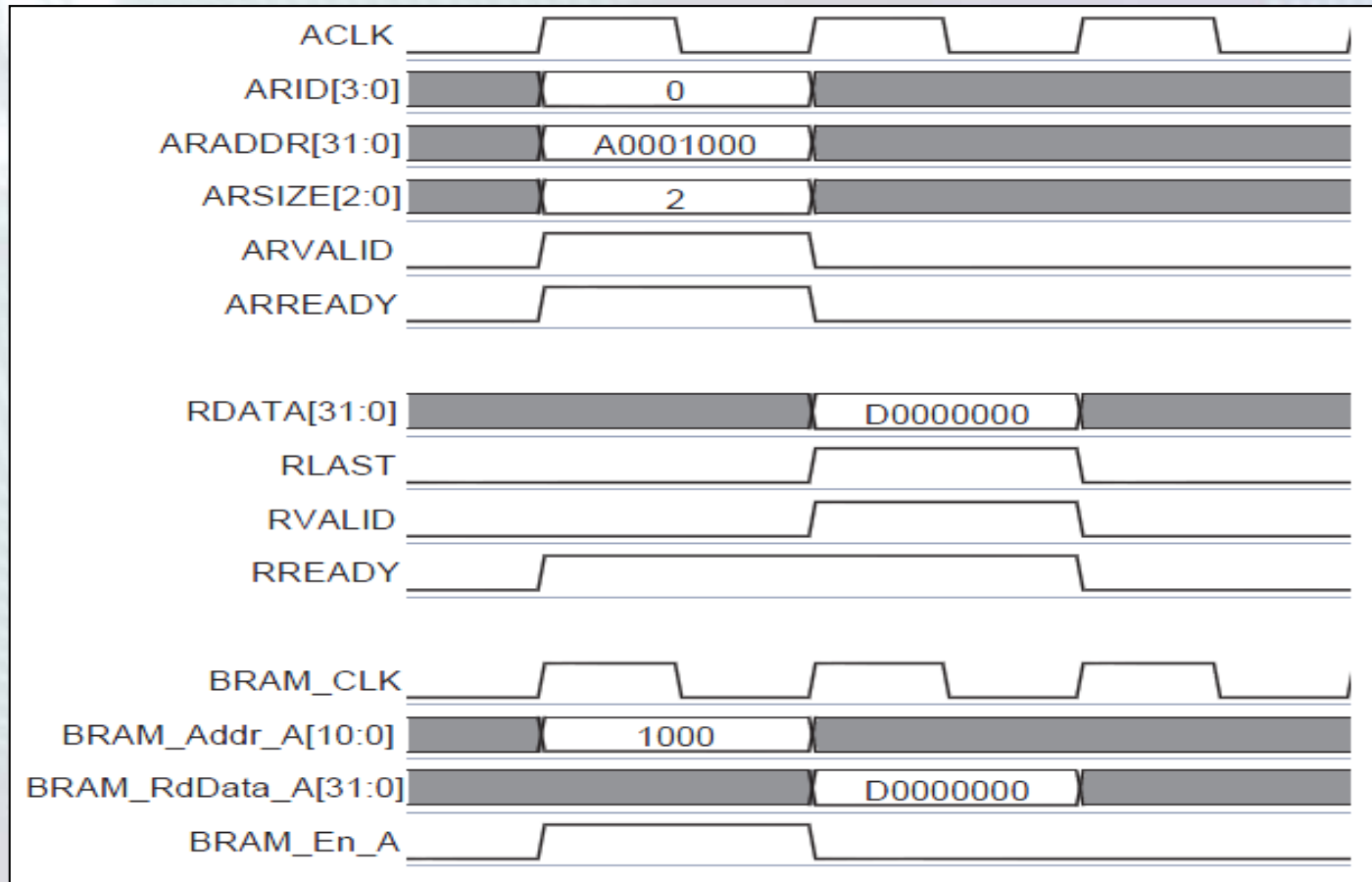
AXI – Jelek (olvasási adat csatorna)

- ***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): kizárólagos adatátvitel OK
 - SLVERR (10): slave hiba
 - DECERR (11): címdekódolási hiba

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

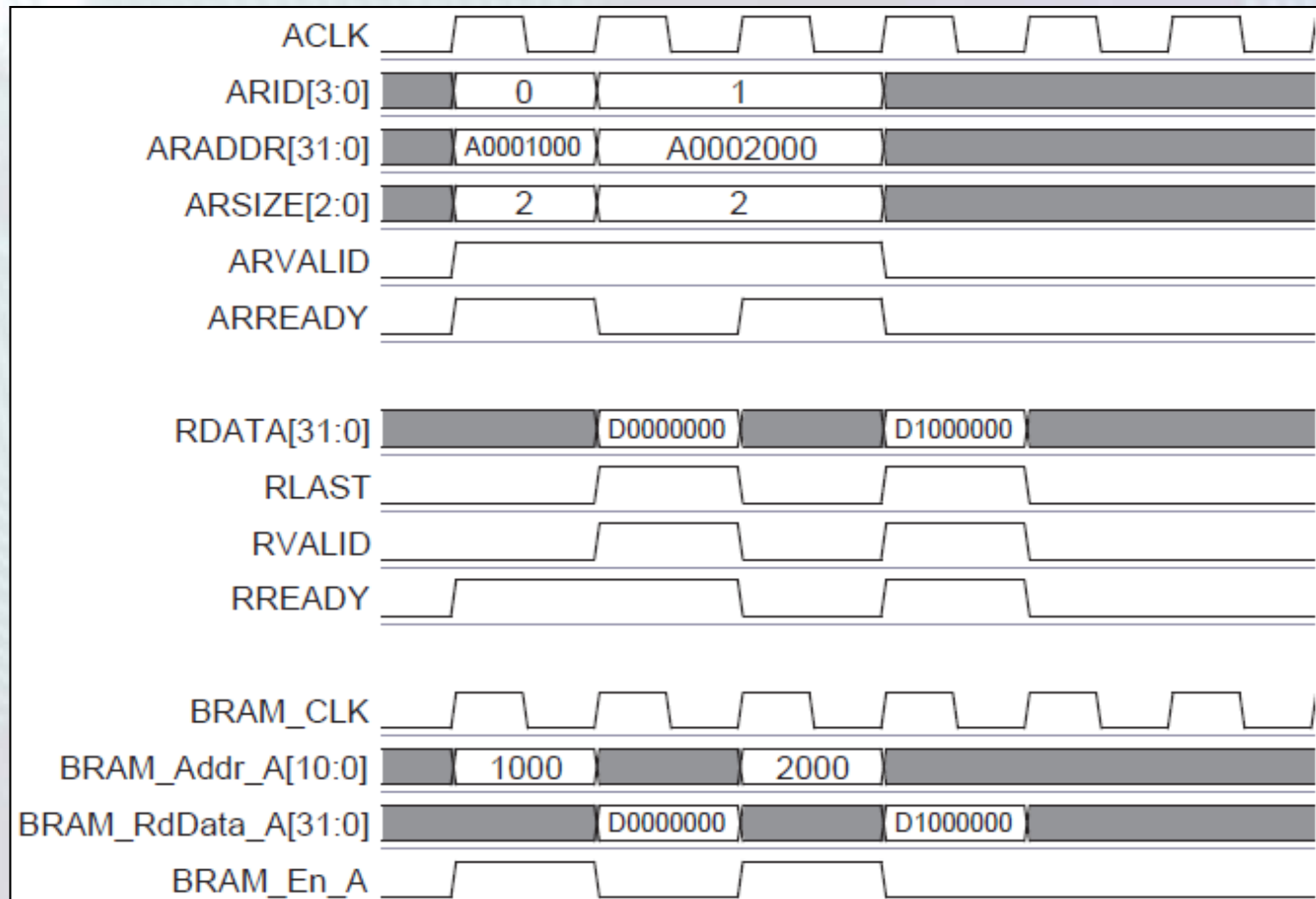
AXI – Tranzakció példák

AXI4 BRAM vezérlő: egyszeres olvasás



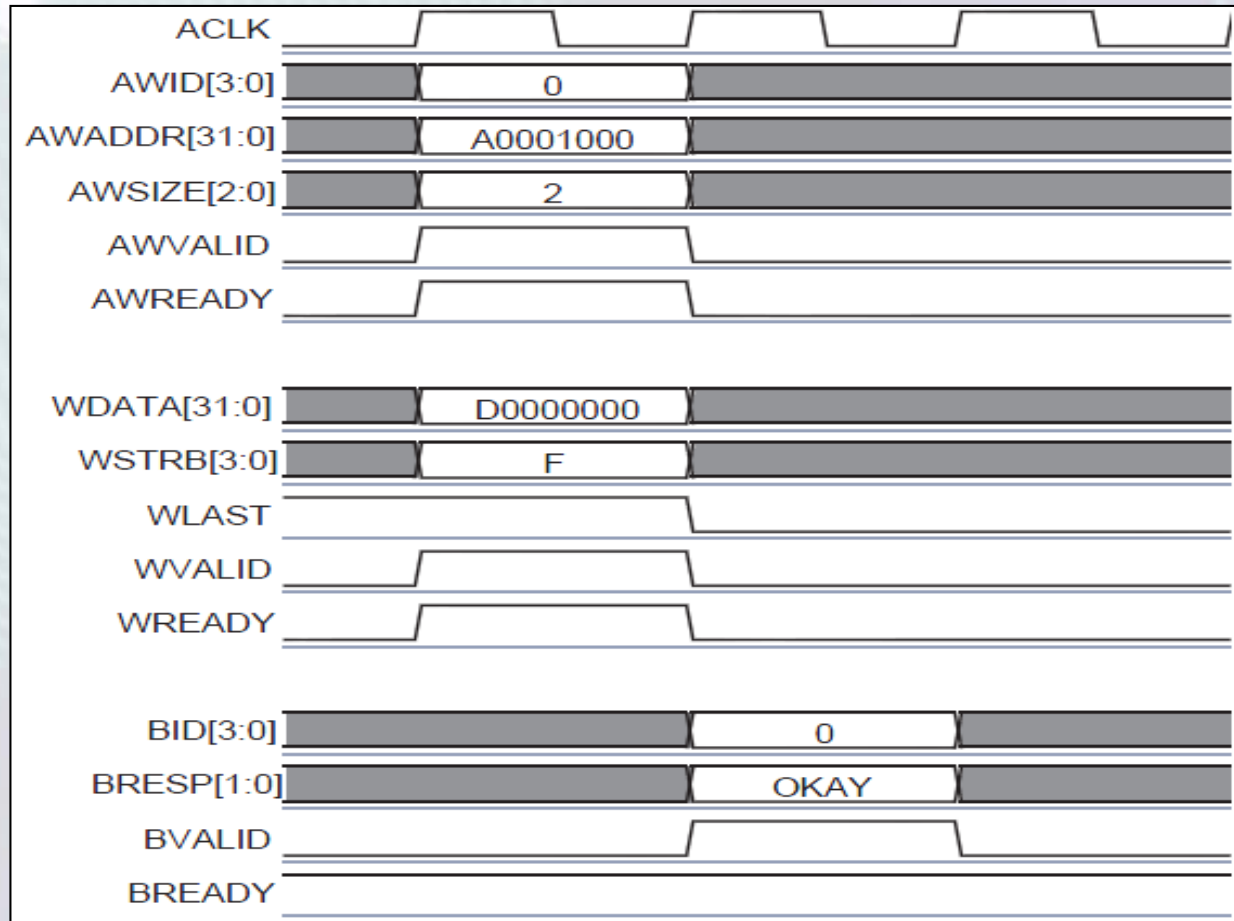
AXI – Tranzakció példák

AXI4 BRAM vezérlő: több olvasás egymás után



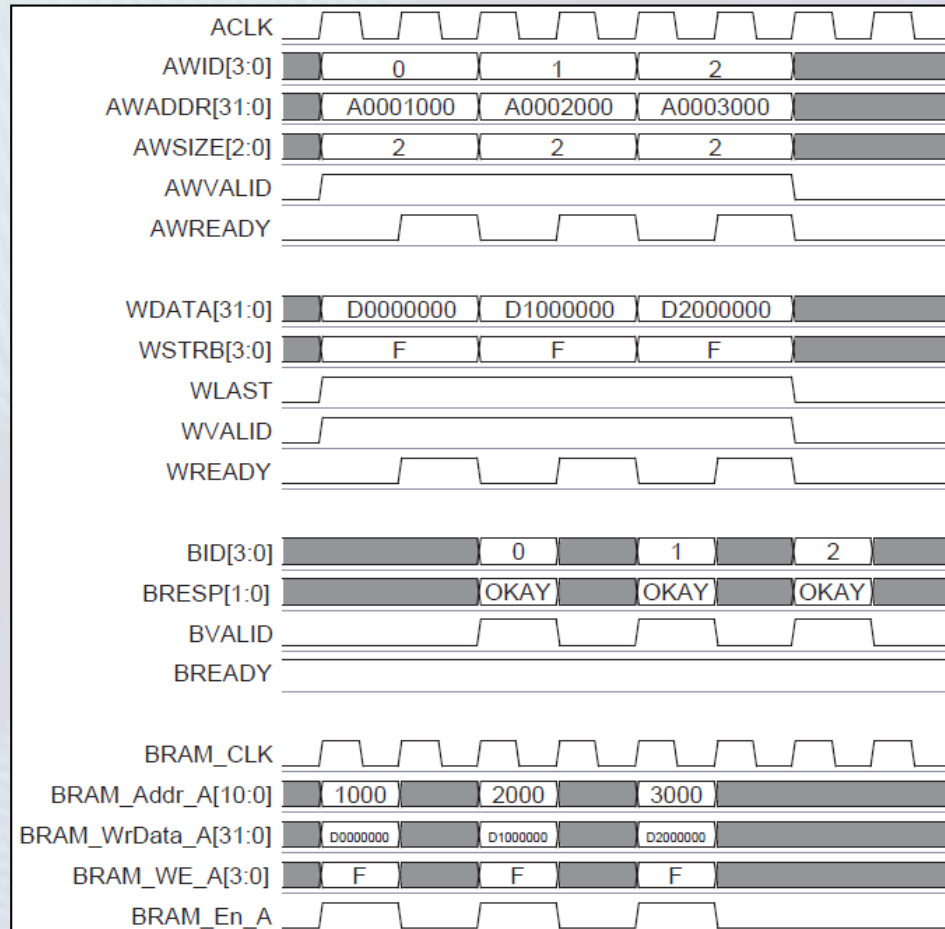
AXI – Tranzakció példák

AXI4 BRAM vezérlő: egyszeres írás



AXI – Tranzakció példák

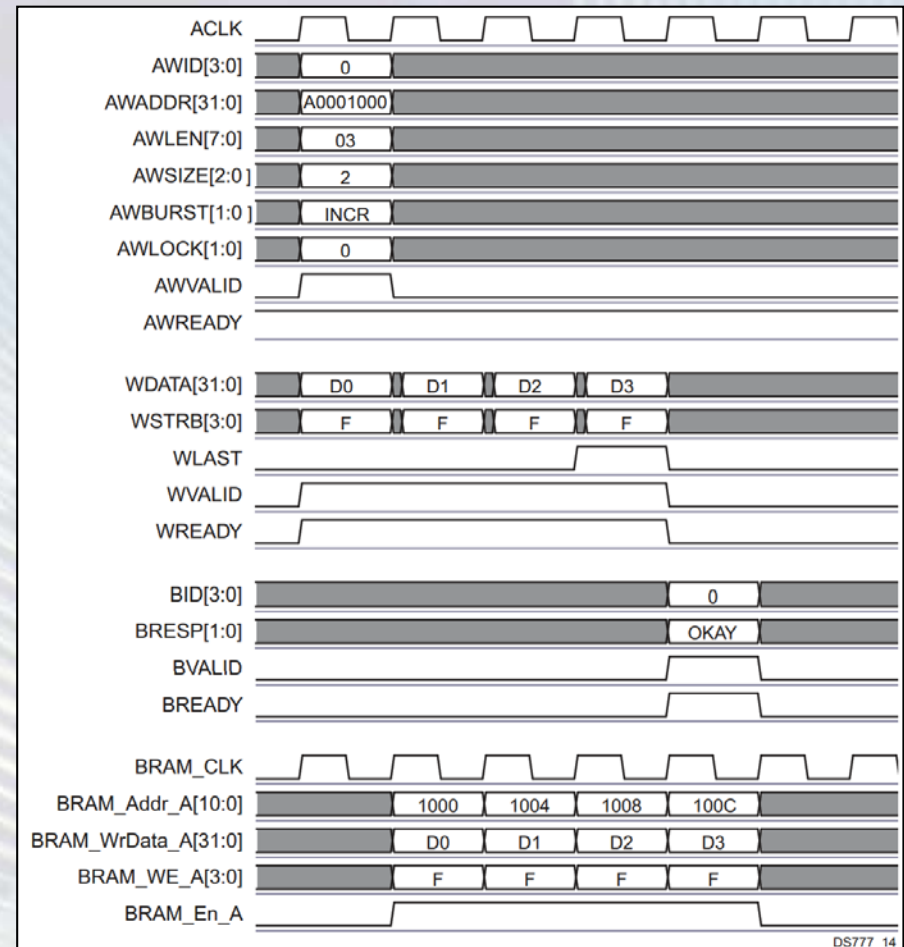
AXI4 BRAM vezérlő: több írás egymás után



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

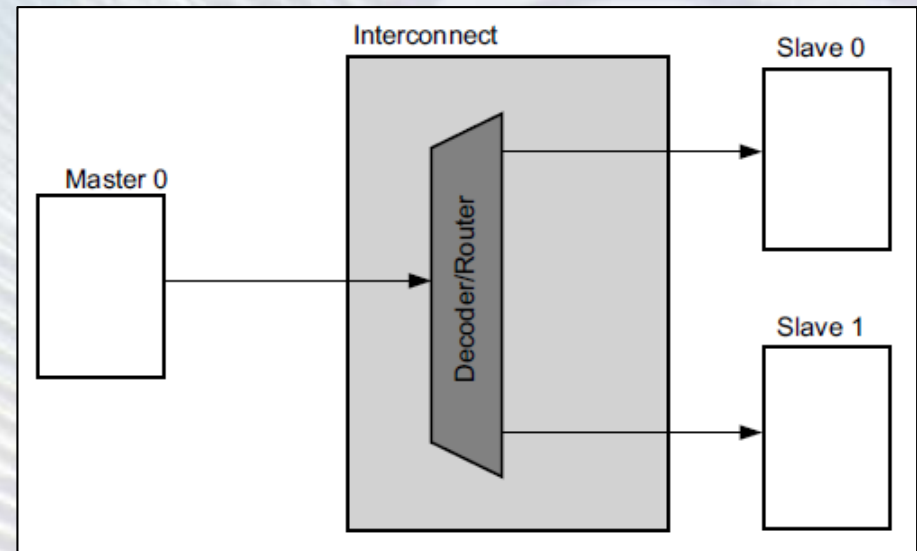
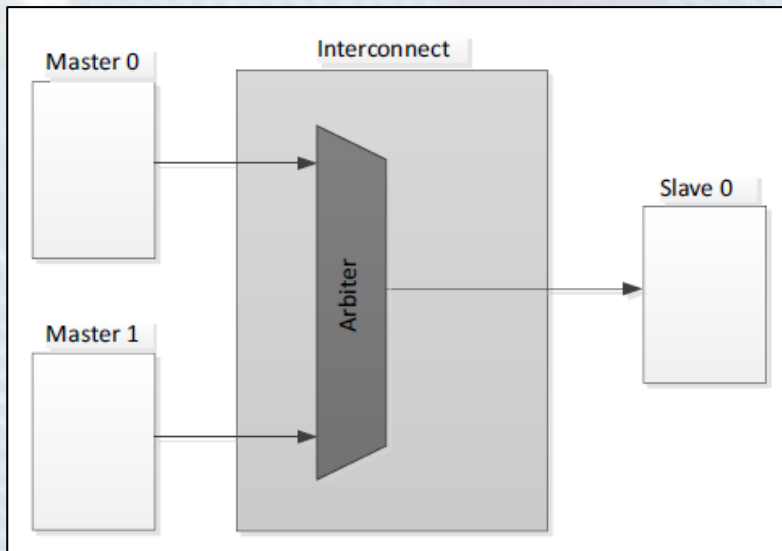
Szolgáltatások:

- **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

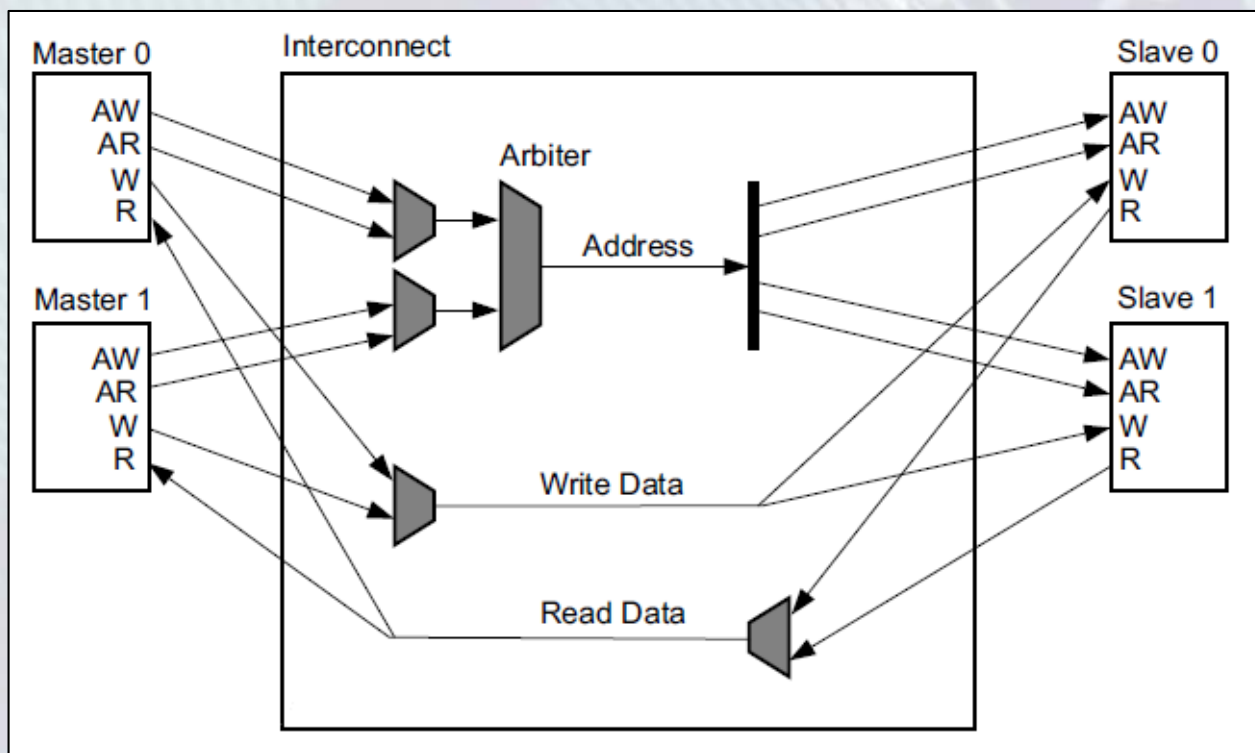
- N master, 1 slave: egyszerű arbiter
- 1 master, N slave: egyszerű dekóder, nincs arbitráció



AXI Interconnect, AXI SmartConnect

Felhasználási esetek

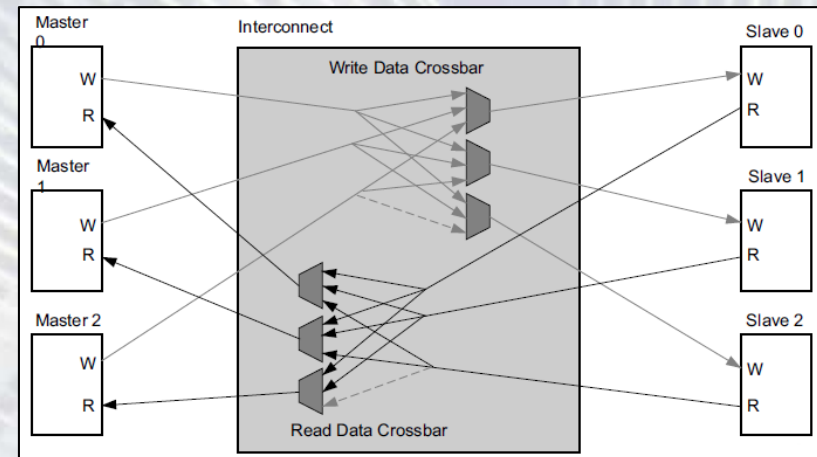
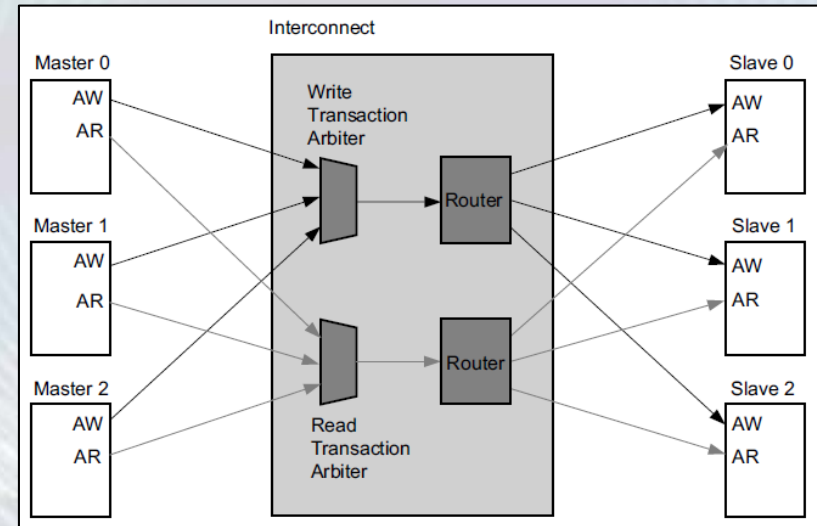
- N master, M slave, osztott elérési mód (egy időben csak egy tranzakció lehet aktív)



AXI Interconnect, AXI SmartConnect

Felhasználási esetek

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

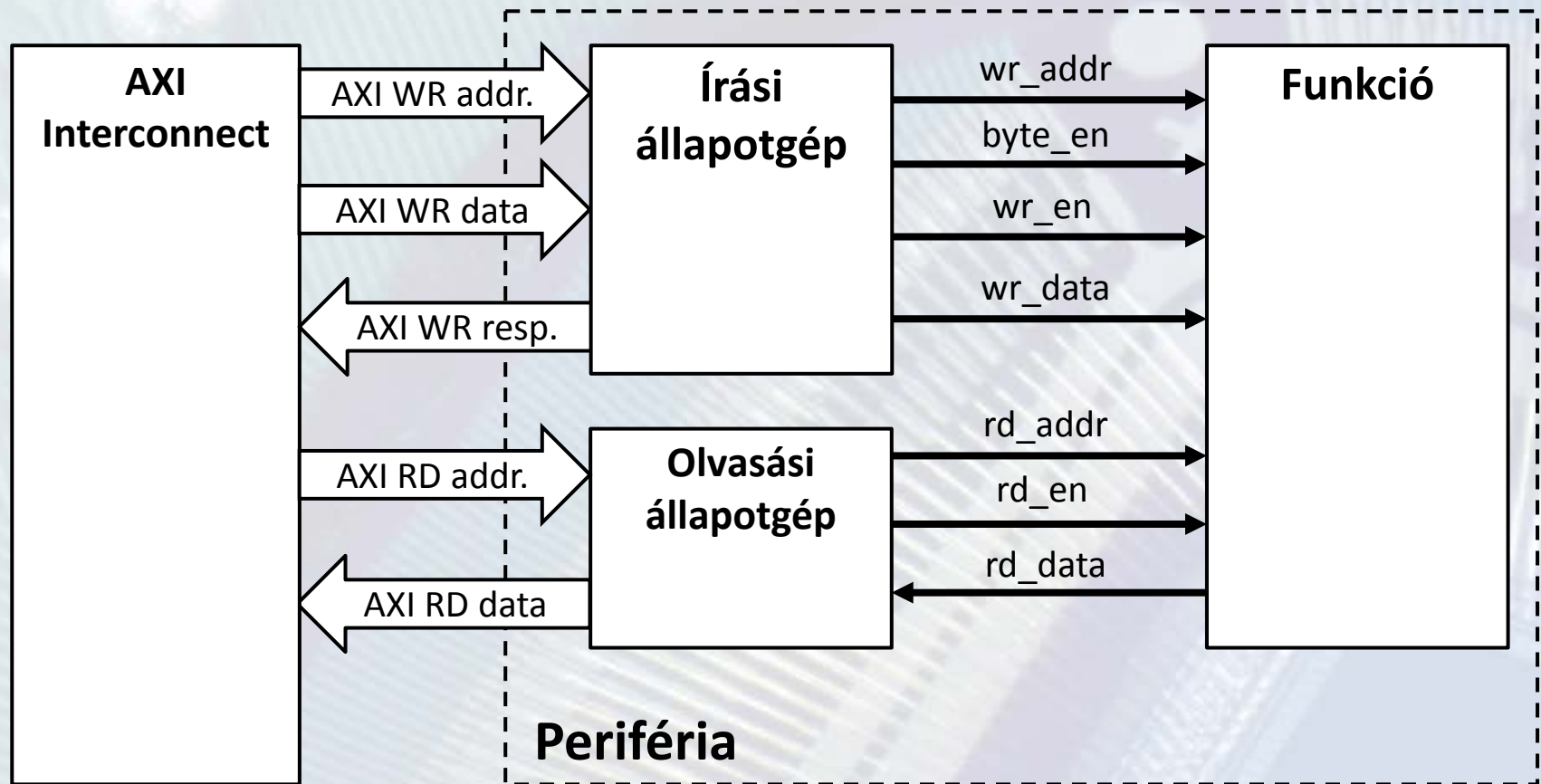


Példa: AXI4-Lite slave interfész

- Csak egy ütemből álló tranzakciók
- 32 bites írási és olvasási adatbusz
- Az írási és olvasási tranzakciók függetlenek
 - *Írási állapotgép*: cím vétele → adat vétele → periféria regiszter írása → írás nyugtázása
 - *Olvasási állapotgép*: cím vétele → periféria regiszter olvasása → az adat elküldése
- Nem a leghatékonyabb, de könnyen megérthető
- A handshake szabályok és függőségek biztosan teljesülnek ez esetben

Példa: AXI4-Lite slave interfész

Blokkvázlat

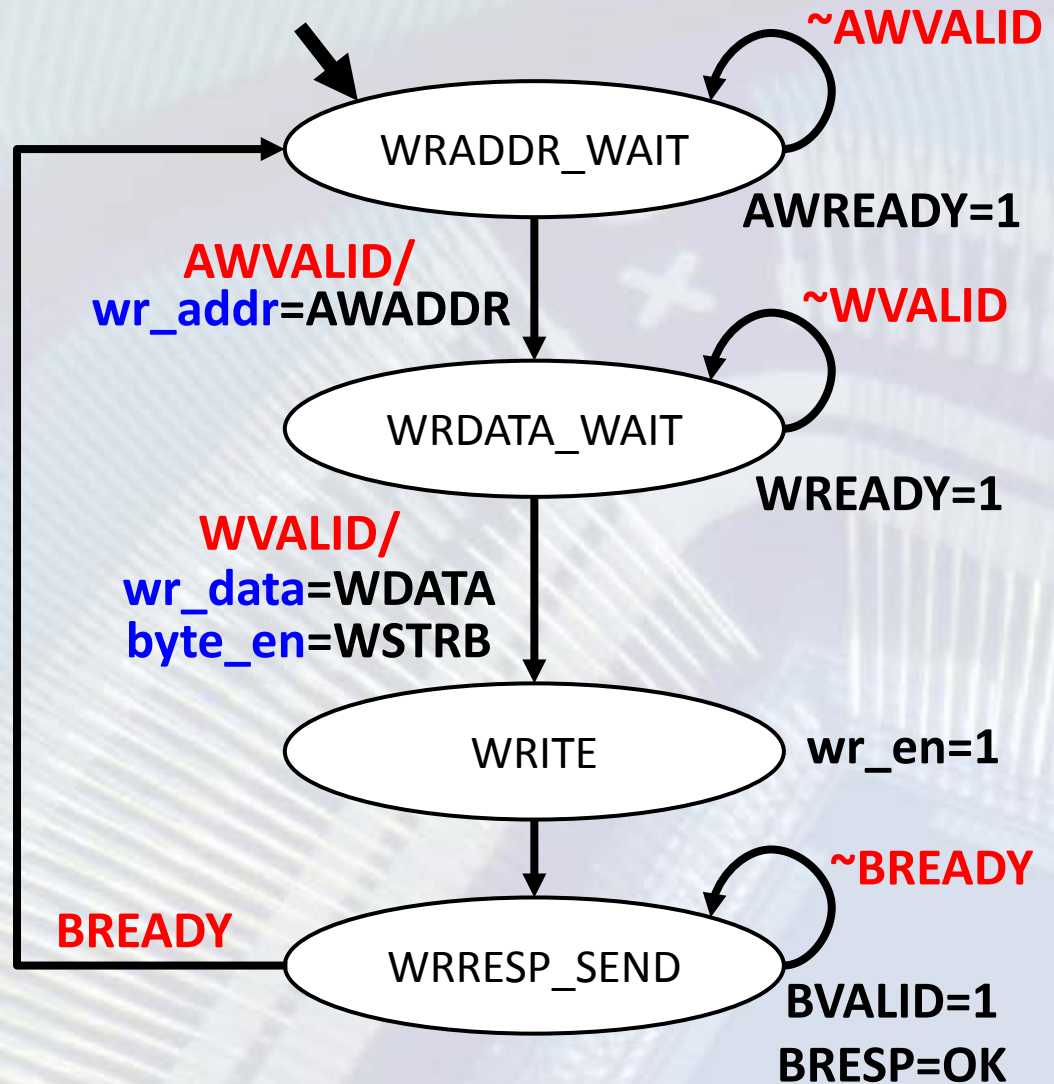


Példa: AXI4-Lite slave interfész

Írási állapotgép

Regiszterek

- wr_addr: N bit
- wr_data: 32 bit
- byte_en: 4 bit

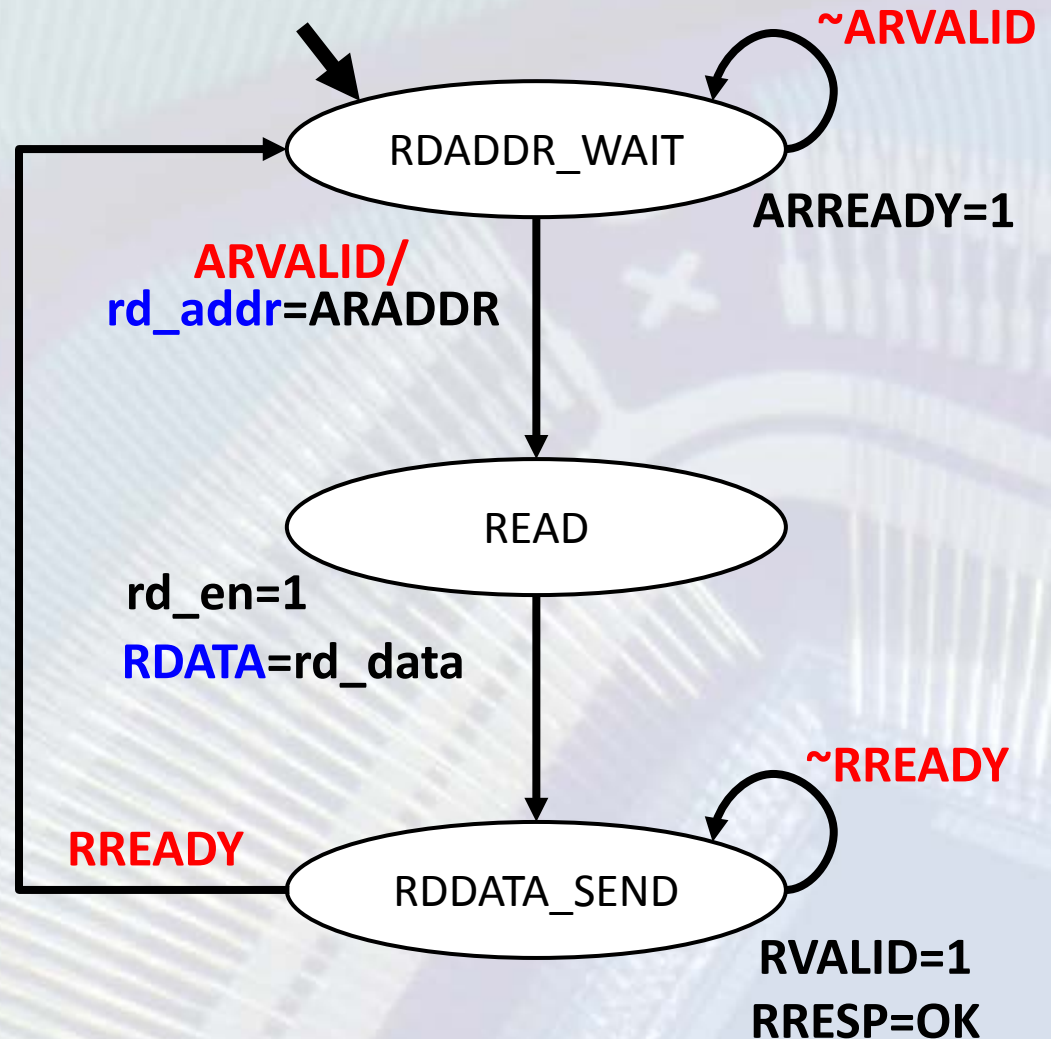


Példa: AXI4-Lite slave interfész

Olvasási állapotgép

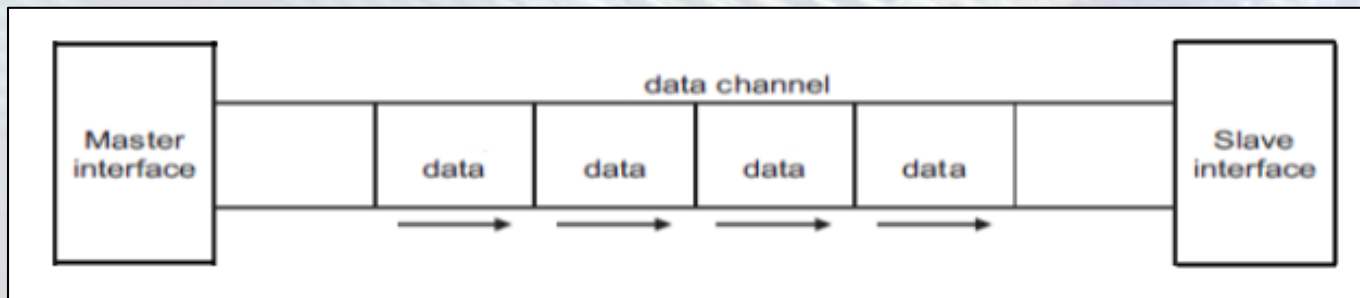
Regiszterek

- rd_addr: N bit
- RDATA: 32 bit



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)**
- **Ritka, folyamatos, igazított és nem igazított folyamatok**
- **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**

AXI Stream – Adatfolyam példák

1. Bájtfolyam: csak adatbájtok és null bájtok
2. Folyamatos, igazított folyam: csak adatbájtok a csomagokban
3. Folyamatos, nem igazított folyam: csak adatbájtok és pozíció bájtok a csomagok elején és/vagy végén
4. Ritka folyam: csak adatbájtok és pozíció bájtok

Null	Null	D-07	D-0A	Null	D-0F
D-01	Null	D-06	D-09	Null	D-0E
Null	D-03	D-05	D-08	D-0C	Null
D-00	D-02	D-04	Null	D-0B	D-0D

1

D-03	D-07	D-0B	D-0F	D-13
D-02	D-06	D-0A	D-0E	D-12
D-01	D-05	D-09	D-0D	D-11
D-00	D-04	D-08	D-0C	D-10

2

D-00	D-04	D-08	D-0C	D-10	Position
Position	D-03	D-07	D-0B	D-0F	Position
Position	D-02	D-06	D-0A	D-0E	Position
Position	D-01	D-05	D-09	D-0D	D-11

3

D-03	D-07	Position	D-0F	D-13
Position	D-06	D-0A	Position	D-12
D-01	Position	D-09	D-0D	Position
D-00	D-04	Position	D-0C	D-10

4

Xilinx AXI Stream IP-k

- **Részletek: AXI4-Stream Infrastructure IP Suite (PG085)**
- **AXI4-Stream Broadcaster**
 - A bemeneti master folyamat megismétli több (2 – 16) kimeneti slave folyamaton
- **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 interfészek illesztését biztosítja
- **AXI4-Stream Switch**
 - Útválasztást biztosít master és slave egységek között
- **AXI4-Stream Interconnect**
 - A Broadcaster és a Combiner kivételével a fenti IP-eket foglalja egyetlen konfigurálható egységbe