

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

# **Rendszerarchitektúrák labor**

## **Xilinx EDK**

**Raikovich Tamás**

**BME MIT**

# Labor tematika (Xilinx EDK)

- **1. labor:**
  - A Xilinx EDK fejlesztői környezet ismertetése
- **2. labor:**
  - Egyszerű processzoros rendszer összeállítása
  - Egyszerű szoftver alkalmazások készítése
- **3. labor:**
  - Saját periféria illesztése
  - Megszakításkezelés
  - Egyidejű HW/SW fejlesztés (debugger, ChipScope)

# Témakörök

- Beágyazott rendszerek
- *MicroBlaze processzor*
- EDK alapok
- Gyári és saját IP-k hozzáadása
- Szoftverfejlesztés
- HW és SW együttes fejlesztése

# A MicroBlaze processzor

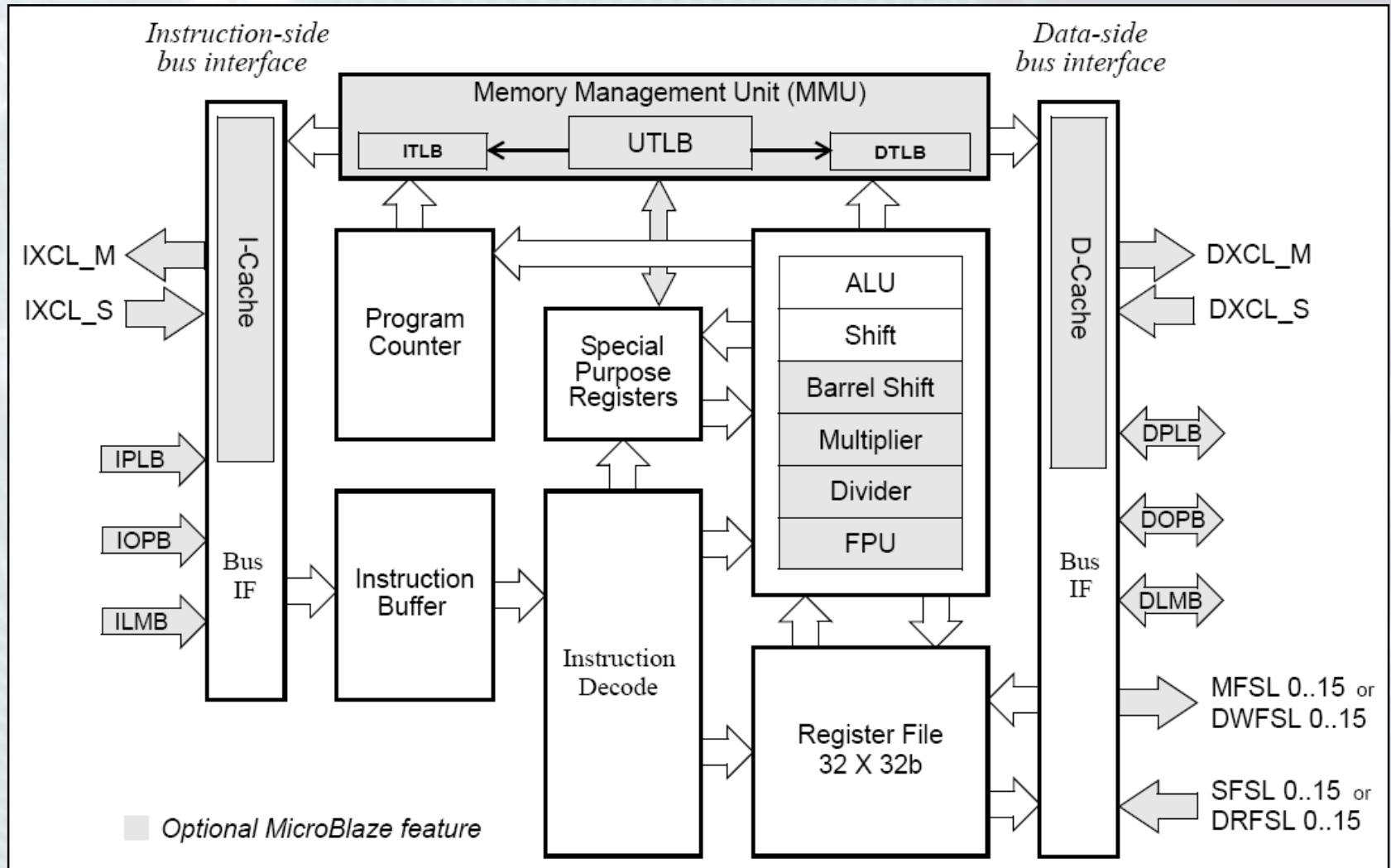
- **32 bites általános célú RISC processzor**
  - Alkalmas önálló feladatok végrehajtására
  - Kiegészítő egység a nagyteljesítményű PowerPC-s rendszerekben → egységes interfész felület
- **Lágy processzor mag**
  - Az FPGA programozható erőforrásaiból épül fel
- **Rugalmas rendszerkialakíthatóság**
  - Minimális verzió lokális belső memóriával
  - Nagy rendszer külső memóriával, sok perifériával

# A MicroBlaze processzor

## A MicroBlaze processzor jellemzői:

- **Harvard architektúra**
  - Szétválasztott utasítás és adat oldali busz interfész
- **32 bites címbusz és adatbusz**
- **32 általános célú regiszter**
- **Ortogonalis utasításkészlet**
  - 32 bites utasítások
  - 3 operandus, 2 címzési mód
- **Egyszeres kibocsátású pipeline**
  - Kisebb erőforrás igényű változat: 3 fokozatú pipeline
  - Nagyteljesítményű változat: 5 fokozatú pipeline
- **Nagymértékben konfigurálható**

# A MicroBlaze processor



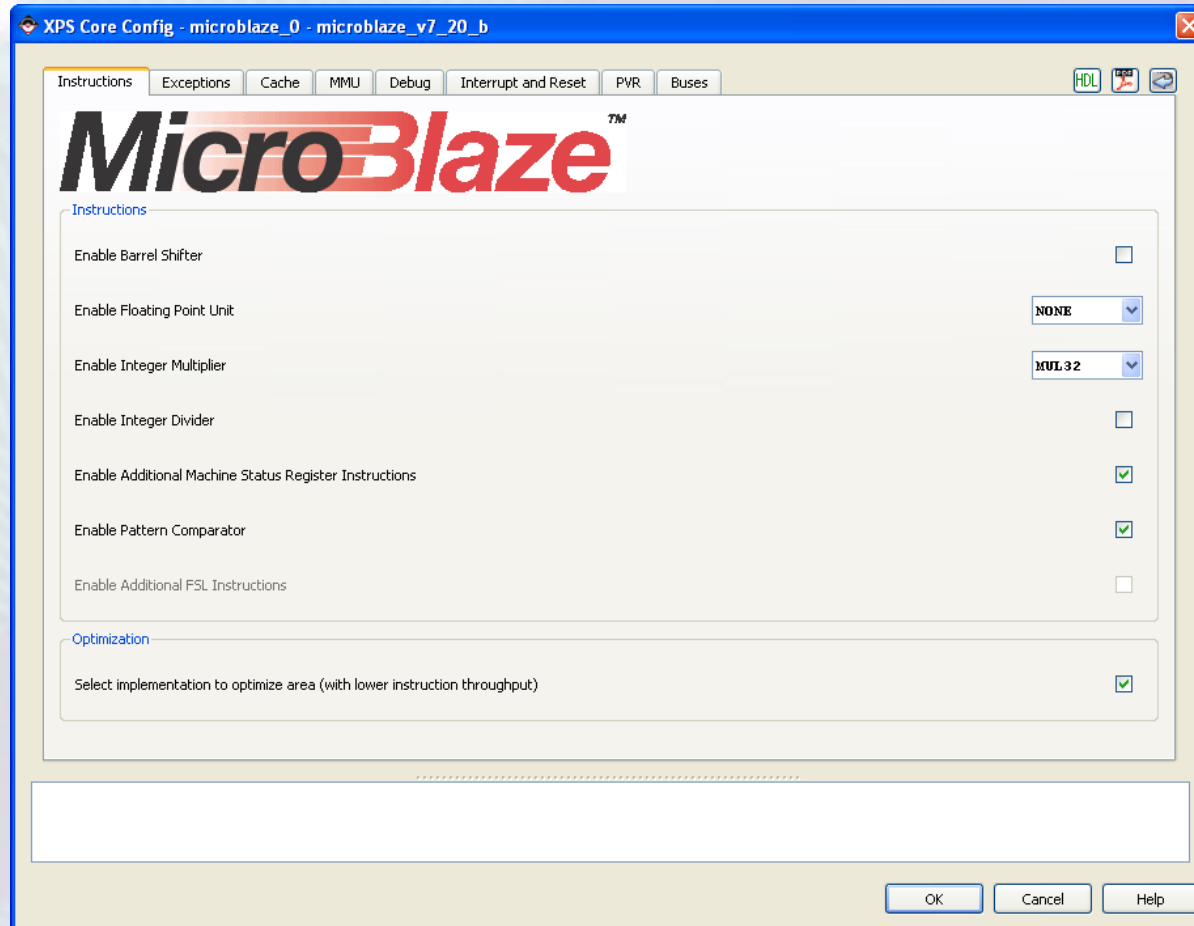
# A MicroBlaze processzor

## A MicroBlaze főbb konfigurációs beállításai:

- **Utastítás és adat oldali busz interfészek**
  - Processor Local Bus (PLB), On-chip Peripheral Bus (OPB)
  - Local Memory Bus (LMB), Fast Simplex Link (FSL), CacheLink
- **Utastítás és adat cache**
- **MMU (Linux operációs rendszer futtatásához)**
- **Opcionális utastítások**
  - Szorzás (hardveres szorzó), osztás (hardveres osztó)
  - Barrel shifter, pattern compare
  - Lebegőpontos műveletek támogatása (FPU)
- **Hardveres debug modul**
- **Hardveres kivételek támogatása**
- **Pipeline fokozatok száma**

# A MicroBlaze processzor

## A MicroBlaze főbb konfigurációs beállításai:





# A MicroBlaze processzor

**A PowerPC és a MicroBlaze processzorok az IBM CoreConnect buszrendszerrel használják:**

- **Az IBM saját fejlesztésű, belső buszrendszere**
- **A specifikáció nyilvános, licenstdíj nélkül használható**
- **Teljes, de egyúttal komplex megoldás**
- **Hierarchikus buszrendszer**
  - PLB: Processor Local Bus
  - OPB: On-chip Peripheral Bus
  - DCR: Device Control Register bus
- **Egy kissé a szokásos teljes számítógépes struktúrát modellezi**

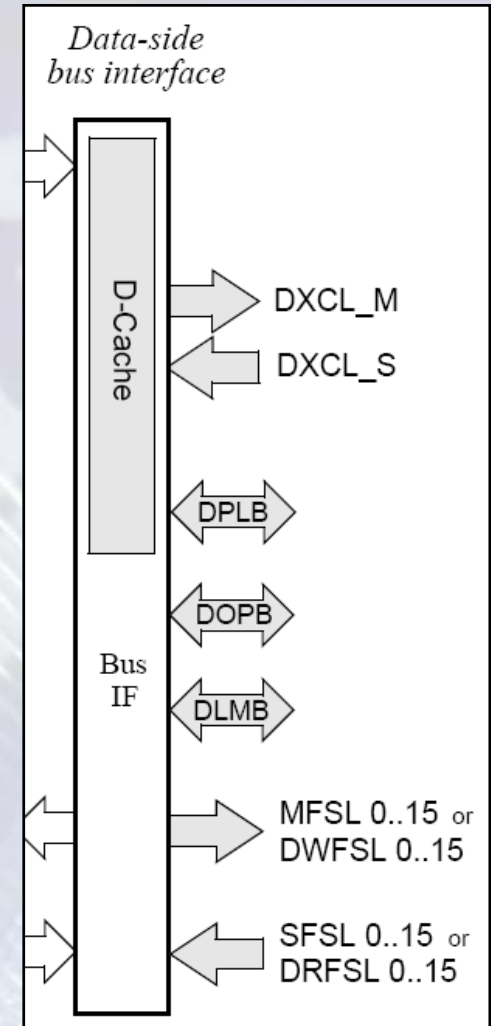
# A MicroBlaze processzor

- **Processor Local Bus (PLB)**
  - A processzor, a memóriák és a nagy sebességű blokkos átvitelű perifériák összekapcsolásához
- **On-chip Peripheral Bus (OPB)**
  - A szokásos, kisebb teljesítményű perifériákhoz
  - Megfelelő sávszélesség, blokkos adatátvitel is
  - Multi-master üzemmód lehetséges
  - Nagy címtartomány, egyszerű struktúra
- **Device Control Register bus (DCR)**
  - Speciális vezérlőregiszter hozzáférés
  - A többi busz forgalmát így ezzel nem zavarjuk

# A MicroBlaze processzor

## A MicroBlaze processzor busz interfészei:

- Külön utasítás és adat oldali busz interfészek
- Processor Local Bus (PLB)
  - Az EDK 9.2 verziótól kezdve az OPB helyett
  - A gyári perifériák is a PLB-re kapcsolódnak
- On-chip Peripheral Bus (OPB)
  - A kompatibilitás megőrzése miatt
- Local Memory Bus (LMB)
  - A belső Blokk-RAM-ok illesztéséhez
- Fast Simplex Link (FSL)
  - FIFO alapú interfész saját funkciók illesztéséhez
  - Processzorok közötti kommunikáció
- Xilinx CacheLink (XCL)
  - Cache és memóriaverérlő közötti kommunikáció
  - Lényegében FSL + egyedi protokoll



# A MicroBlaze processzor

## A MicroBlaze processzor adattípusai:

- Bájt (8 bit), félszó (16 bit) és szó (32 bit) adattípusok
- Big-Endian: MSB a kisebb, LSB a nagyobb bájtcímen
- Bit fordított: MSb sorszáma mindig 0

Table 1-2: Word Data Type

Byte address	n	n+1	n+2	n+3
Byte label	0	1	2	3
Byte significance	MSByte			LSByte
Bit label	0			31
Bit significance	MSBit			LSBit

Table 1-3: Half Word Data Type

Byte address	n	n+1
Byte label	0	1
Byte significance	MSByte	LSByte
Bit label	0	15
Bit significance	MSBit	LSBit

Table 1-4: Byte Data Type

Byte address	n	
Bit label	0	7
Bit significance	MSBit	LSBit

# A MicroBlaze processzor (LMB)

## A Local Memory Bus (LMB):

- Szinkron busz a belső Blokk-RAM-ok nagysebességű eléréséhez
- Minimális számú vezérlőjel
- Egyszerű protokoll
- Megjegyzés: saját periféria is illeszthető az LMB-re
  - Ehhez a *microblaze\_v2\_1\_0.tcl* fájlt módosítani kell

```
# To allow custom LMB IP please change this variable to 1  
# set IGNORE_CUSTOM_LMB_IP_ERROR 0  
set IGNORE_CUSTOM_LMB_IP_ERROR 1
```

- A saját periféria ne lassítsa le a rendszert

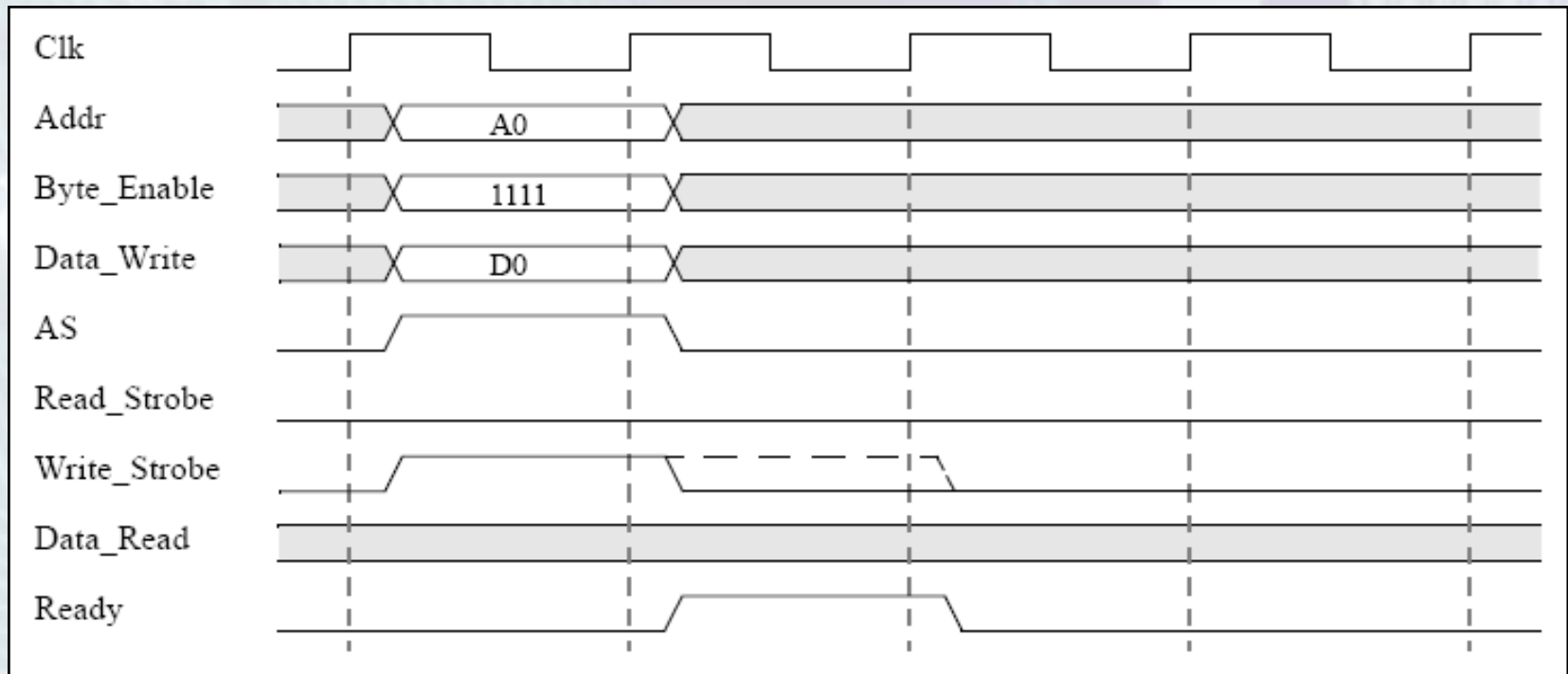
# A MicroBlaze processzor (LMB)

## A Local Memory Bus (LMB) vonalai:

Signal	Data Interface	Instruction Interface	Type	Description
Addr[0:31]	Data_Addr[0:31]	Instr_Addr[0:31]	O	Address bus
Byte_Enable[0:3]	Byte_Enable[0:3]	<i>not used</i>	O	Byte enables
Data_Write[0:31]	Data_Write[0:31]	<i>not used</i>	O	Write data bus
AS	D_AS	I_AS	O	Address strobe
Read_Strobe	Read_Strobe	IFetch	O	Read in progress
Write_Strobe	Write_Strobe	<i>not used</i>	O	Write in progress
Data_Read[0:31]	Data_Read[0:31]	Instr[0:31]	I	Read data bus
Ready	DReady	IReady	I	Ready for next transfer
Clk	Clk	Clk	I	Bus clock

# A MicroBlaze processzor (LMB)

## Az LMB elemi írási ciklusa:



# A MicroBlaze processzor (LMB)

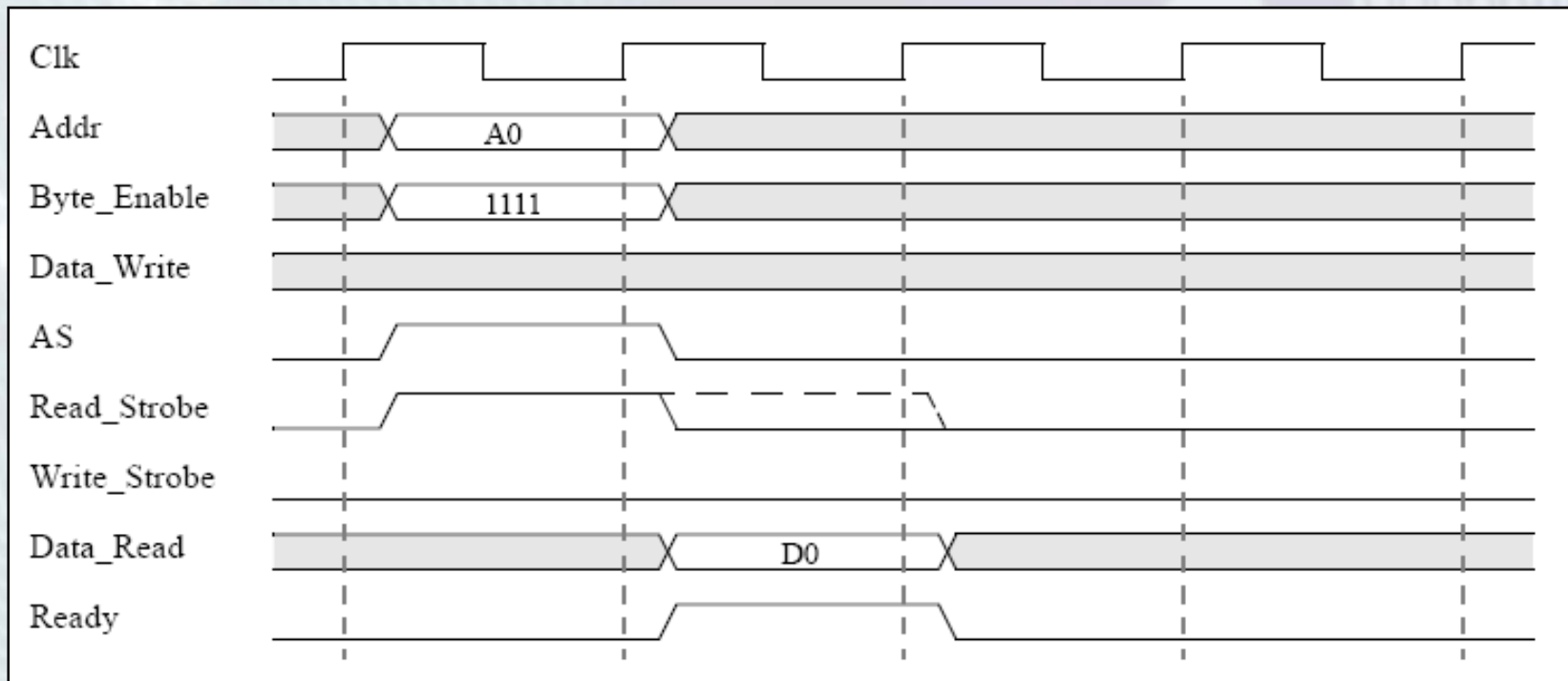
## Az LMB elemi írási ciklusa: 2 órajel

- **1. ciklus:**
  - Cím kiadása és érvényesítése az **AS** jellel
  - Adat LMB-nél: bájt engedélyező jelek is (**Byte\_Enable**)
  - Adat kiadása és az írás jelzése a **Write\_Strobe** jellel
- **2. ciklus:**
  - A Blokk-RAM a **Clk** felfutó élére tárolja az adatot
  - A **Ready** jellel jelezzük a művelet befejeződését
  - A ciklus után a vezérlőjelek inaktívak lesznek



# A MicroBlaze processzor (LMB)

Az LMB elemi olvasási ciklusa:



# A MicroBlaze processzor (LMB)

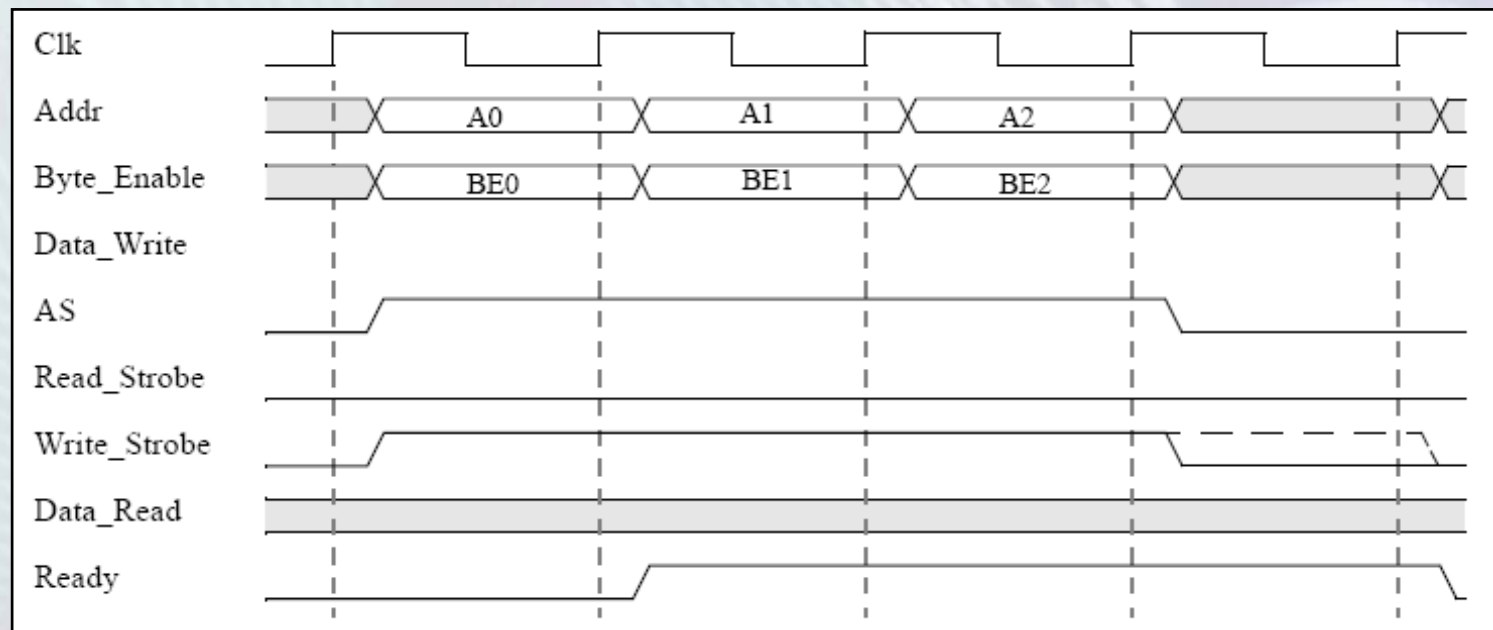
## Az LMB elemi olvasási ciklusa: 2 órajel

- **1. ciklus:**
  - Cím kiadása és érvényesítése az **AS** jellel
  - Adat LMB-nél: bájt engedélyező jelek is (**Byte\_Enable**)
  - Az olvasás jelzése a **Read\_Strobe** jellel
- **2. ciklus:**
  - A Blokk-RAM a **Clk** felfutó élére mintavételezi a parancsot, majd kiadja az érvényes adatot
  - A **Ready** jellel jelezzük a művelet befejeződését
  - A ciklus után a vezérlőjelek inaktívak lesznek

# A MicroBlaze processzor (LMB)

## LMB többszörös írási hozzáférés:

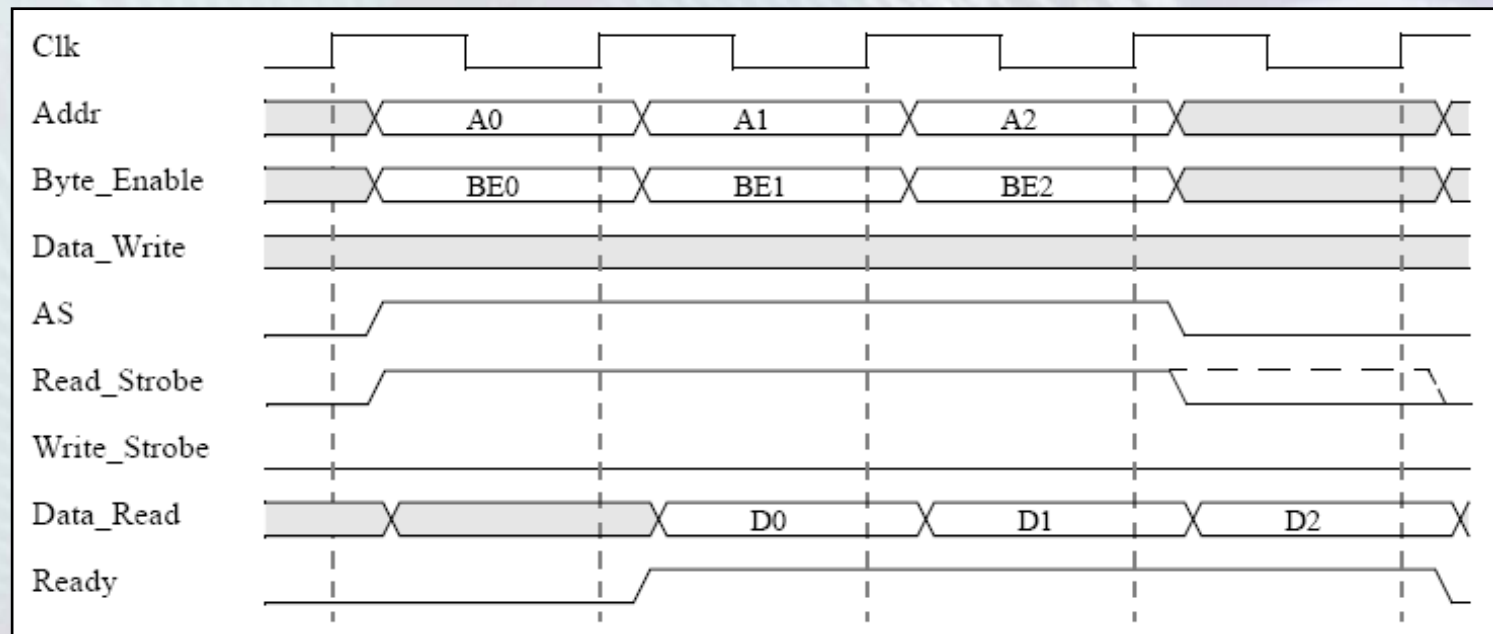
- Új adat minden ciklusban
- Az *AS*, a *Write\_Strobe* és a *Ready* jelek folyamatosan aktívak
- A cím és a *Byte\_Enable* minden ciklusban tetszőleges lehet



# A MicroBlaze processzor (LMB)

## LMB többszörös olvasási hozzáférés:

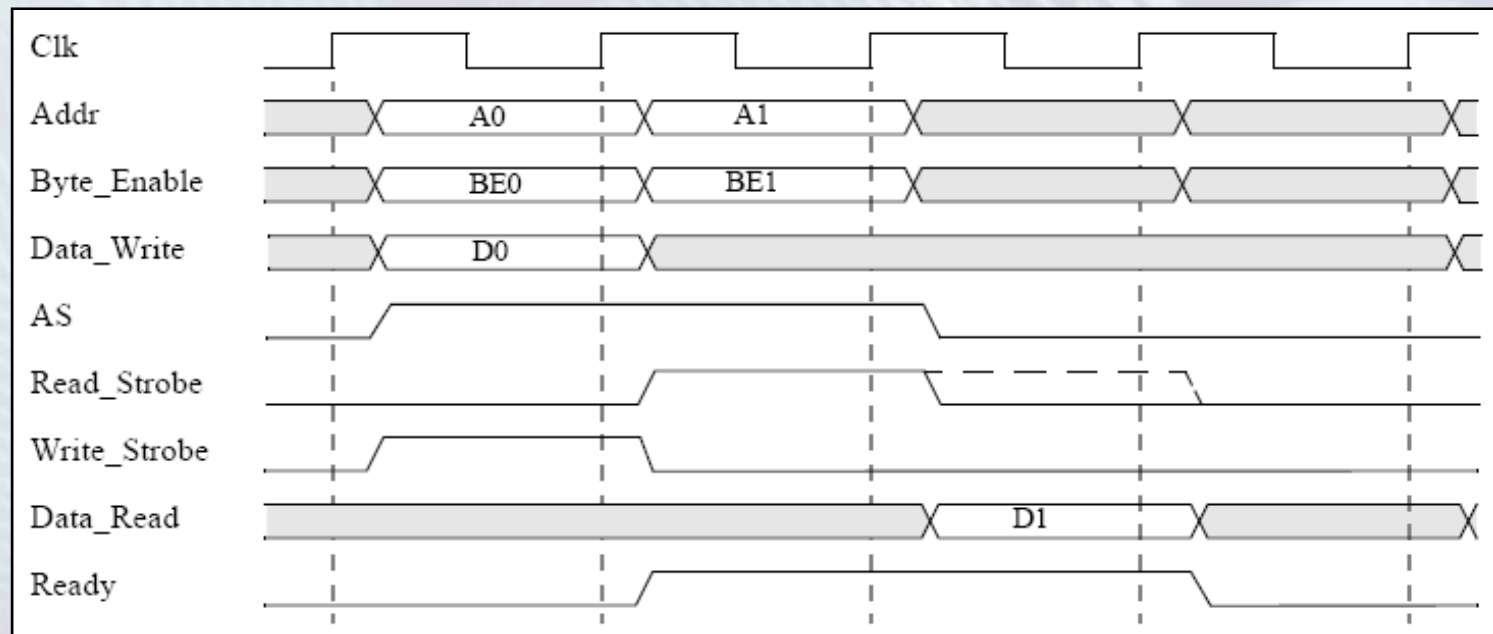
- Új adat minden ciklusban
- Az *AS*, a *Read\_Strobe* és a *Ready* jelek folyamatosan aktívak
- A cím minden ciklusban új, tetszőleges lehet



# A MicroBlaze processzor (LMB)

## LMB váltott írási-olvasási hozzáférés:

- Hasonló mint a többszörös írási vagy olvasási hozzáférés
- Írás esetén a *Write\_Strobe* jel aktív
- Olvasás esetén a *Read\_Strobe* jel aktív



# A MicroBlaze processzor (LMB)

## LMB bájt kezelés írásnál: nincs átrendezés

Address [30:31]	Byte_Enable [0:3]	Transfer Size	Write Data Bus Bytes			
			Byte0	Byte1	Byte2	Byte3
11	0001	byte				rD[24:31]
10	0010	byte			rD[24:31]	
01	0100	byte		rD[24:31]		
00	1000	byte	rD[24:31]			
10	0011	halfword			rD[16:23]	rD[24:31]
00	1100	halfword	rD[16:23]	rD[24:31]		
00	1111	word	rD[0:7]	rD[8:15]	rD[16:23]	rD[24:31]

# A MicroBlaze processzor (LMB)

## LMB bájt kezelés olvasásnál:

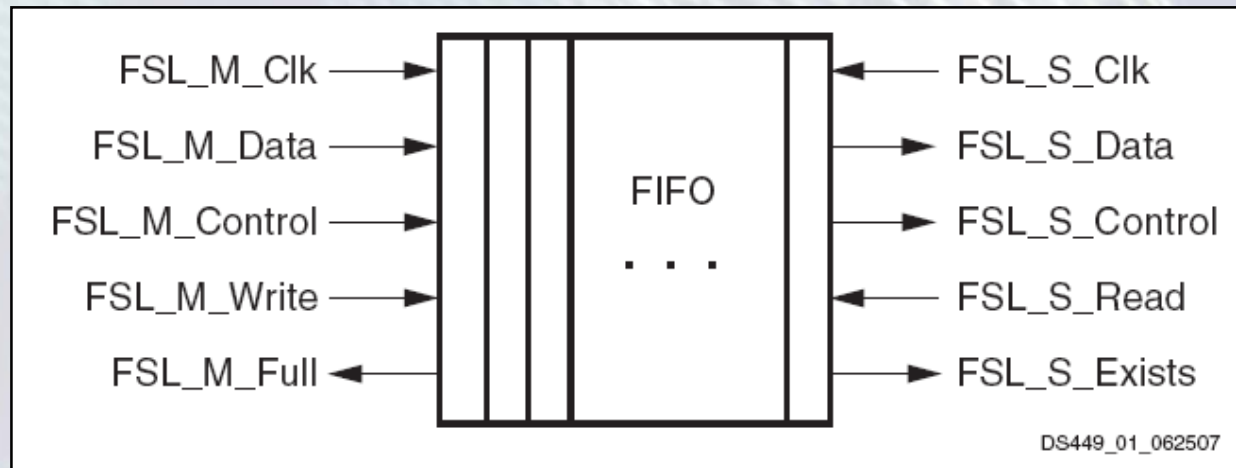
- Jobbra rendezés (a MicroBlaze végzi el)
- Nem támogatott a szavasnál (32 bit) nagyobb átvitel (mert nincs dinamikus buszméret kezelés)

Address [30:31]	Byte_Enable [0:3]	Transfer Size	Register rD Data			
			rD[0:7]	rD[8:15]	rD[16:23]	rD[24:31]
11	0001	byte				Byte3
10	0010	byte				Byte2
01	0100	byte				Byte1
00	1000	byte				Byte0
10	0011	halfword			Byte2	Byte3
00	1100	halfword			Byte0	Byte1
00	1111	word	Byte0	Byte1	Byte2	Byte3

# A MicroBlaze processzor (FSL)

## Fast Simplex Link (FSL):

- FIFO alapú, egyirányú pont-pont kommunikációs csatorna
- Nagy sebességű adatátvitelt biztosít
- A master és a slave oldal órajele eltérő lehet (aszinkron FIFO)
- Dedikált MicroBlaze utasítások az FSL adatátvitelhez
- Felhasználási példák:
  - Saját funkció illesztése a processzorhoz („utasítás kiterjesztés”)
  - Processzorok közötti kommunikációs interfész





# A MicroBlaze processzor (FSL)

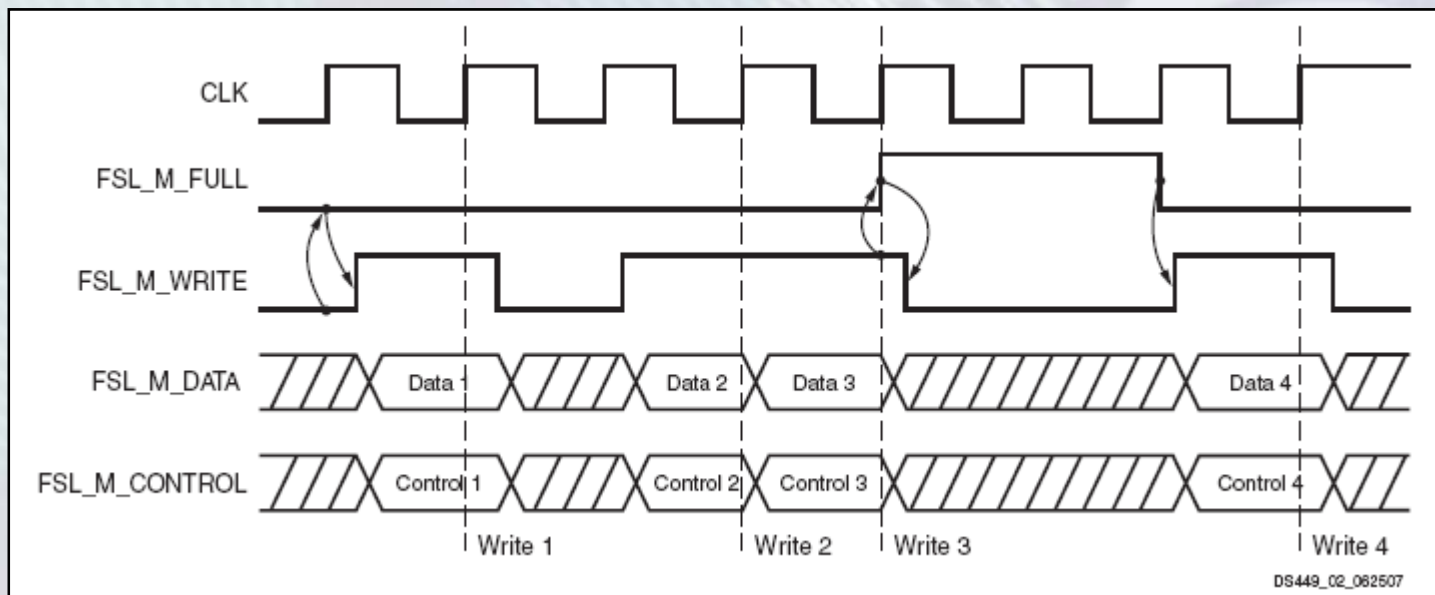
## Fast Simplex Link (FSL):

- **Master interfész: a FIFO írásához**
  - *FSL\_M\_Clk*: órajel
  - *FSL\_M\_Data*: 32 bites adat
  - *FSL\_M\_Control*: 1 bites adat (az adott protokoll definiálja)
  - *FSL\_M\_Write*: FIFO írás engedélyező jel
  - *FSL\_M\_Full*: FIFO tele jelzés
- **Slave interfész: a FIFO olvasásához**
  - *FSL\_S\_Clk*: órajel
  - *FSL\_S\_Data*: 32 bites adat
  - *FSL\_S\_Control*: 1 bites adat (az adott protokoll definiálja)
  - *FSL\_S\_Read*: FIFO olvasás jel
  - *FSL\_S\_Exists*: Érvényes adat van a FIFO-ban (a FIFO nem üres)

# A MicroBlaze processzor (FSL)

## FSL írási művelet:

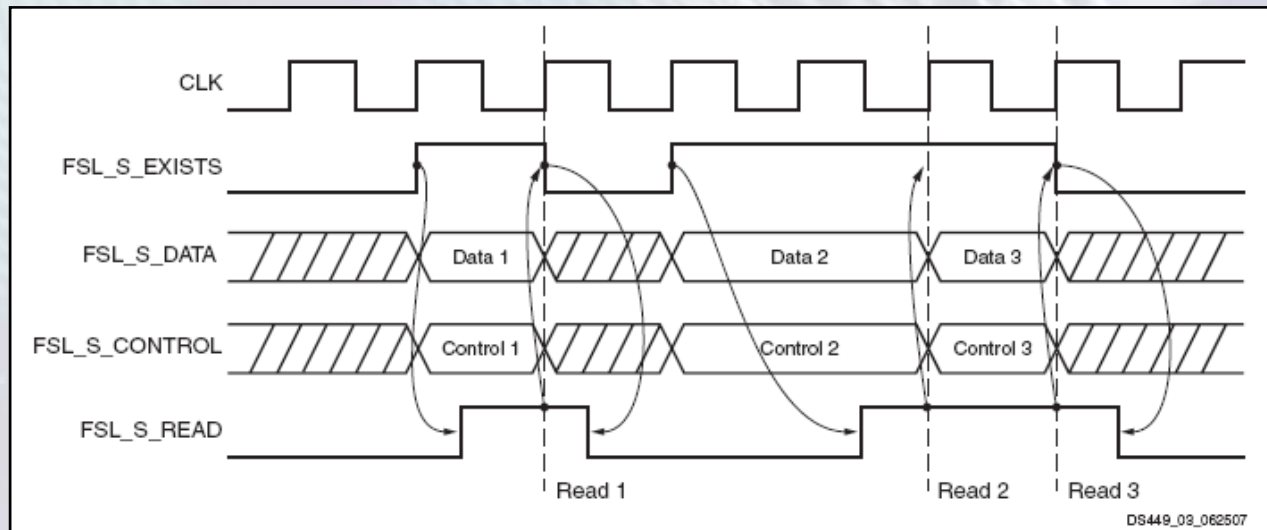
- Ha az *FSL\_M\_WRITE* aktív, akkor a *CLK* felfutó élére az *FSL\_M\_DATA* és az *FSL\_M\_CONTROL* beíródik a FIFO-ba
- Ha az *FSL\_M\_FULL* aktív (FIFO tele jelzés), akkor nem szabad írni a FIFO-ba



# A MicroBlaze processzor (FSL)

## FSL olvasási művelet:

- Ha az *FSL\_S\_EXISTS* aktív, akkor érvényes adat van az *FSL\_S\_DATA* és az *FSL\_S\_CONTROL* vonalakon
- Ha az *FSL\_S\_READ* aktív, akkor a *CLK* felfutó élére a FIFO kiadja a következő érvényes adatot
- Ha a FIFO kiürült, akkor *FSL\_S\_EXISTS* inaktívvá válik
- Ha az *FSL\_S\_EXISTS* inaktív, akkor nem szabad olvasni a FIFO-ból



# A MicroBlaze processzor (PLB)

## Processor Local Bus (PLB):

- Nagysebességű kapcsolatot biztosít a master és a slave eszközök között
- Teljesen szinkron működés, egyetlen órajel
- Centralizált busz hozzáférés vezérlés: PLB arbiter
- 32 bites, 64 bites vagy 128 bites adatbusz
- Nagyteljesítményű, kis késleltetésű, rugalmas adatátvitel
  - Szétválasztott címbusz, írási és olvasási adatbusz megosztott adatátviteli ciklus képességgel
  - Konkurens írási és olvasási ciklusok, max. 2 átvitel/órajel
  - Cím pipeline a késleltetés csökkentése érdekében
  - Arbitráció végrehajtása az aktuális adatátvitel alatt

# A MicroBlaze processzor (PLB)

## Processor Local Bus (PLB):

- Legfeljebb 16 master egység támogatott
- Legfeljebb 16 slave egység támogatott
- Xilinx implementáció:
  - Az eredeti PLB egyszerűsített változata
  - Nem támogat minden adatátviteli módot
- **Bonyolult, a részletekbe most nem megyünk bele**
- **Xilinx IP Interface (IPIF) modul**
  - Kétirányú interfész a PLB busz és a saját IP között
  - Egyszerűsített interfészt biztosít a saját IP felé

# A MicroBlaze processzor (pipeline)

## Pipeline utasítás végrehajtás:

- Egyszeres kibocsátású pipeline: egyszerre egy utasítás lehívása történik
- 3 fokozatú pipeline: a kisebb erőforrás igényű változatban
  - Fetch, Decode, Execute
- 5 fokozatú pipeline: a nagyobb teljesítményű változatban
  - Fetch (IF), Decode (OF), Execute (EX)
  - Access Memory (MEM), Writeback (WB)

	cycle 1	cycle 2	cycle 3	cycle4	cycle5	cycle6	cycle7
instruction 1	Fetch	Decode	Execute				
instruction 2		Fetch	Decode	Execute	Execute	Execute	
instruction 3			Fetch	Decode	Stall	Stall	Execute

	cycle 1	cycle 2	cycle 3	cycle 4	cycle 5	cycle 6	cycle 7	cycle 8	cycle 9
instruction 1	IF	OF	EX	MEM	WB				
instruction 2		IF	OF	EX	MEM	MEM	MEM	WB	
instruction 3			IF	OF	EX	Stall	Stall	MEM	WB

# A MicroBlaze processzor (pipeline)

## A szokásos pipeline kérdések:

- **Elágazások a programban**
  - Ugrási utasítások
  - Szubrutinhívás
  - Megszakításkezelés, kivételkezelés
- **Következmény**
  - A lehívott és dekódolt utasítást nem kell végrehajtani
  - Pipeline kiürítés és újratöltés: ciklusok vesznek el
- **A hatékonyság javítása: delay slot**
  - A processzor végrehajtja a már lehívott utasításokat
  - Az ugrás csak ezután következik be
  - Igazából programszervezés kérdése

# A MicroBlaze processzor (regiszterek)

## Regiszterkészlet:

- 32 darab 32 bites általános célú regiszter
- Speciális regiszterek
- HW megkötések
- C/C++ programok esetén használati konvenciók

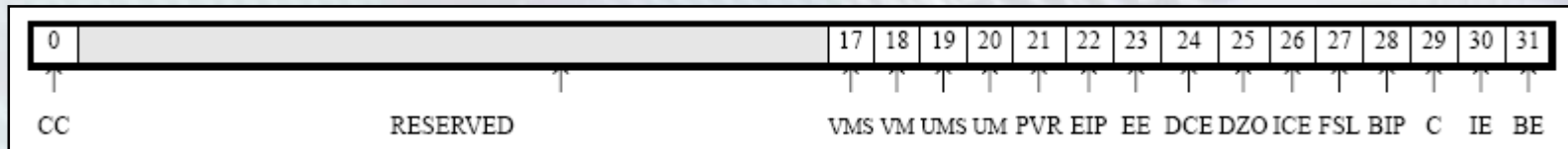
Register	Type	Enforcement	Purpose
R0	Dedicated	HW	Value 0
R1	Dedicated	SW	Stack Pointer
R2	Dedicated	SW	Read-only small data area anchor
R3-R4	Volatile	SW	Return Values/Temporaries
R5-R10	Volatile	SW	Passing parameters/Temporaries
R11-R12	Volatile	SW	Temporaries
R13	Dedicated	SW	Read-write small data area anchor
R14	Dedicated	HW	Return address for Interrupt
R15	Dedicated	SW	Return address for Sub-routine
R16	Dedicated	HW	Return address for Trap (Debugger)
R17	Dedicated	HW, if configured to support HW exceptions, else SW	Return address for Exceptions
R18	Dedicated	SW	Reserved for Assembler/Compiler Temporaries
R19	Non-volatile	SW	Must be saved across function calls. Callee-save
R20	Dedicated or Non-volatile	SW	Reserved for storing a pointer to the Global Offset Table (GOT) in Position Independent Code (PIC). Non-volatile in non-PIC code. Must be saved across function calls. Callee-save
R21-R31	Non-volatile	SW	Must be saved across function calls. Callee-save



# A MicroBlaze processzor (regiszterek)

## Regiszterkészlet: speciális regiszterek

- Írásukhoz és olvasásukhoz külön utasítások
- RPC: programszámláló (32 bites, csak olvasható)
- MSR: státusz regiszter



- Regiszterek a kivételkezeléshez: EAR, ESR, BTR, EDR
- FSR: lebegőpontos státusz regiszter
- MMU regiszterek: PID, ZPR, TLBLO, TLBHI, TLBX, TLBSX
- PVR: processzor verzió regiszter (PVR0 – PVR10)

# A MicroBlaze processzor (utasítások)

## MicroBlaze utasításkészlet:

- Viszonylag egyszerű utasításrendszer
- RISC processzor → műveletek csak regiszterben tárolt adatokon
- „A” típusú utasítások:
  - Műveleti kód + cél regiszter + 2 forrás regiszter

Opcode	Destination Reg	Source Reg A	Source Reg B	0	0	0	0	0	0	0	0	0	0	0
0	6	1	1	2									3	
		1	6	1										1

- „B” típusú utasítások:
  - Műveleti kód + cél reg. + 1 forrás reg. + 16 bites konstans

Opcode	Destination Reg	Source Reg A	Immediate Value											
0	6	1	1											3
		1	6											1

- IMM utasítás: 32 bites konstansok használata
- Az IMM és az azt követő utasítás atomi, nem szakíthatók meg

# A MicroBlaze processzor (utasítások)

## MicroBlaze utasításkészlet:

- **A szokásos utasítás csoportok**
  - Aritmetikai utasítások
  - Logikai utasítások
  - Shift utasítások
  - Adatmozgató utasítások
  - Programvezérlési utasítások
  - Konstans méret kiterjesztés (IMM)
  - Lebegőpontos műveletek (FPU használata esetén)
  - Privilegizált utasítások (MMU használata esetén)
- **Assembly nyelven nem fogunk programozni, ezért a részletekbe nem megyünk bele**
- **A teljes utasításkészlet megtalálható a MicroBlaze processzor adatlapjában**

# A MicroBlaze processzor (kivételek)

## Megszakítás- és kivételkezelés:

- Vektortáblázat a 0x0000 – 0x004F címen
- A táblázatban minden eseményhez két szó tartozik
  - IMM + BRI utasítások: tetszőleges 32 bites cím

Event	Vector Address	Register File Return Address
Reset	0x00000000 - 0x00000004	-
User Vector (Exception)	0x00000008 - 0x0000000C	Rx
Interrupt	0x00000010 - 0x00000014	R14
Break: Non-maskable hardware	0x00000018 - 0x0000001C	R16
Break: Hardware		
Break: Software		
Hardware Exception	0x00000020 - 0x00000024	R17 or BTR
Reserved by Xilinx for future use	0x00000028 - 0x0000004F	-

# A MicroBlaze processzor (kivételek)

## Megszakítás és kivételkezelés:

- Egyetlen megszakításkérő bemenet: *Interrupt* port
- Él- vagy szintérzékeny (konfigurálható)
- Ha  $MSR(IE) = 1$  és megszakítás érkezik:
  - $R14 = RPC$ ,  $RPC = 0x0010$  és  $MSR(IE) = 0$
  - Végrehajtódik a megszakításkezelő rutin
  - Visszatérés a megszakításból: RTID utasítás
- A megszakítás rendszer komplexitását a külső megszakítás vezérlő periféria határozza meg