

Dr. Vörösházi Zsolt:

Beágyazott rendszerek fejlesztése (FPGA)



**A felsőfokú informatikai oktatás
minőségének fejlesztése,
modernizációja**

TÁMOP-4.1.2.A/1-11/1-2011-0104



Főkedvezményezett:
Pannon Egyetem
8200 Veszprém
Egyetem u. 10.

Kedvezményezett:
Szegedi Tudományegyetem
6720 Szeged
Dugonics tér 13.



Frissítve: 2019. szeptember 10.

Beágyazott rendszerek fejlesztése (FPGA)

Dr. Vörösházi Zsolt

**2. Beágyazott Rendszer fejlesztő szoftverkörnyezet
(Xilinx Vivado – Embedded Development Kit / Software
Development Kit) bemutatása**



1. Bevezetés – Beágyazott rendszerek, FPGA-k, Digilent ZYBO fejlesztő kártyák és eszközök
2. **Beágyazott Rendszer fejlesztő szoftverkörnyezet (Xilinx Vivado Embedded Development) áttekintése**
3. Beágyazott alap tesztrendszer (BSB - Base System Builder and Board Bring-Up) összeállítása
4. Perifériák hozzáadása (IP adatbázisból) az összeállított beágyazott alaprendszerhez. Saját periféria hozzáadása az összeállított beágyazott alaprendszerhez
5. Szoftver alkalmazások fejlesztése, tesztelése, hibakeresése (debug) Xilinx Vivado SDK (Software Development Kit) használatával

Előzmények:



- Xilinx Vivado Design Suite fejlesztő környezet használatának elsajátítása
- Alapszintű C/C++ tudás
 - Programozás I. / II.
- Digitális Technika / Számítógép Architektúrák alapismerete

Xilinx termék portfólió (2019)



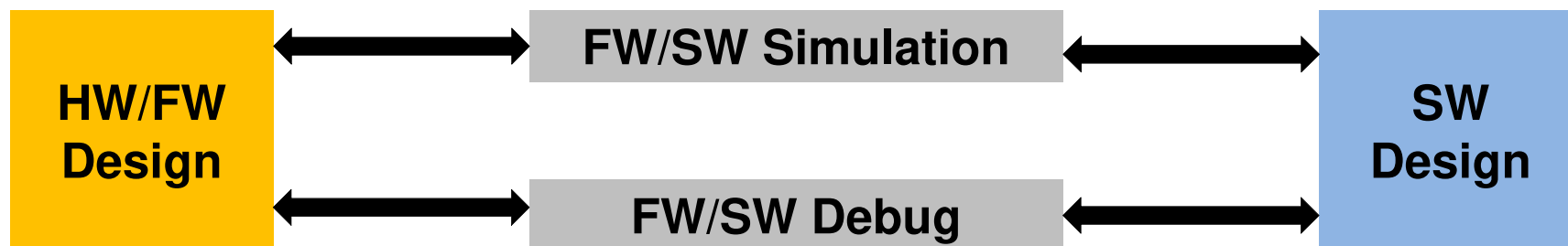
- Vivado
 - Vivado Design Suite (2015.4-ig)
 - **Vivado Design Suite HLx 2018.3** (UltraScale SoC-k, és 7-es sorozatok támogatása). Jelenleg: 2019.1 sp3
- Vivado HLS: High Level Synthesis
 - C/C++, OpenCV támogatás
- SDx: Software Defined
 - SDAccel, SDSoC (reVision), SDNet
 - Cloud támogatás, C,C++, OpenCL
- Korábban: ISE, és XPS Platform Studio
 - FPGA 6-os sorozatok, ill. régebbi eszközök támogatása. Támogatása mára megszűnt!

Laborokon a **Xilinx Vivado Design Suite HLx 2018.3** integrált fejlesztő környezetet használjuk!

- Vivado Project Navigator: keretrendszer, integrált fejlesztő környezet, amelyből más program modulok indíthatók (korábbi projekt management szemléletmódú PlanAhead)
 - **Embedded designer / IPI = IP integrátor (~EDK) – Beágyazott firmware rendszer (FPGA logika, és APSoc konfiguráció)**
 - Xilinx IP katalógusból (ingyenes, vagy fizetős)
 - Saját, vagy külső IP
 - Timing Analyzer – Időzítési analízátor
 - FPGA Editor
 - **HW Manager– Bitstream (FPGA konfiguráció) letöltő program**
 - ChipScope: logikai analízátor
- **Vivado SDK – Beágyazott szoftver rendszer (sw alkalmazás)**
 - Eclipse alapú IDE környezet

Keretrendszer fő modul-részei:

- **Embedded Designer / IP integrátor (~EDK):** Xilinx FPGA-alapú integrált fejlesztő környezet Beágyazott Rendszerek összeállításához, paraméterezéséhez (*HW/FW szinten*), gyors prototípus fejlesztésre
- **SDK: Software Developer Kit:** Eclipse-alapú integrált *szoftver* fejlesztő környezet beágyazott processzorok támogatásához (pl. MicroBlaze™, ARM™, stb.):
 - SW: C/C++ nyelvű alkalmazások,
 - GCC, GPP (esetleg külső) fordítók,
 - GNU debugger - hibakeresés
 - Beágyazott OS (Linux disztribúciók) támogatása.



Általános ismertetés

BEÁGYAZOTT RENDSZER-FEJLESZTŐ KÖRNYEZET (XILINX VIVADO)

-  Xilinx Vivado Design Suite fejlesztő eszközök:
 - <https://www.xilinx.com/products/design-tools/vivado.html#documentation>
-  Vivado Getting Started Guide (UG-910)
 - https://www.xilinx.com/support/documentation/sw_manuals/xilinx2019_1/ug910-vivado-getting-started.pdf
-  Zynq-7000 APSoC Concepts, Tools, and Techniques (Hands-on Guide)
 - https://www.xilinx.com/support/documentation/sw_manuals/xilinx2015_3/ug873-zynq-ctt.pdf

Cél:

- Beágyazott rendszerek fejlesztése (firmware) Xilinx FPGA-kra (pl. Spartan-7, Virtex-7, Kintex-7), vagy APSoC-kra (pl. Zynq-7000 sorozat),
- Komponensek (IP) leírása és integrálása, paraméterezése,
- HW/FW tervezés megvalósításának módszerei, lépései,
- eszközök (tools) bemutatása,
- SDK sw-fejlesztő környezet megismerése, hardveres hibakeresés (debug).

Beágyazott rendszer FPGA-n

- Beágyaz(ható/ott) processzoros rendszer fejlesztése:
 - **Xilinx MicroBlaze™** : soft-core processzor, vagy
 - **ARM**: hard-core processzor
 - Periféria összeköttetések:
 - AXI AMBA v.4 busz interfész (Vivado és EDK 12.x –től támogatva)
 - Reset, Clock, Debug portok, kivezetések összekapcsolása
- SW könyvtárak, driverek generálása
 - C/C++ nyelvi támogatás
 - SW alkalmazás összeállítása
 - SW driverek, rutinok (timer, interrupt), előre definiált függvények
- Operációs rendszer használata (OS v. RTOS)
 - Stand-alone (non-OS)
 - Beágyazott Linux OS (PetaLinux), vs. VxWorks RTOS distro-k stb.

A. Beágyazott hardver/firmware fejlesztése

- **IP Integrator: gyors prototípus fejlesztés célhardverre**
- Meglévő beágyazott HW/FW rendszer kibővítése, ha szükséges,
 - IP katalógusban lévő perifériák integrálásával,
 - Egyedi (saját készítésű) IP perifériák segítségével,
- HDL netlista (leíró) generálása

B. Beágyazott SW fejlesztése SDK-ban

- BSP: board support package
- Könyvtárak (library) és driverek generálása
- SW alkalmazás készítése és debug-olása **Software Development Kit (SDK)**
 - Opcionálisan: alkalmazás debug-olása **Xilinx Microprocessor Debug (XMD)** és **GNU debugger (gdb)** segítségével akár ARM/MicroBlaze magokra

C. Implementáció

- Szintézis, MAP, PAR, illetve időzítési analízis (user constraints)

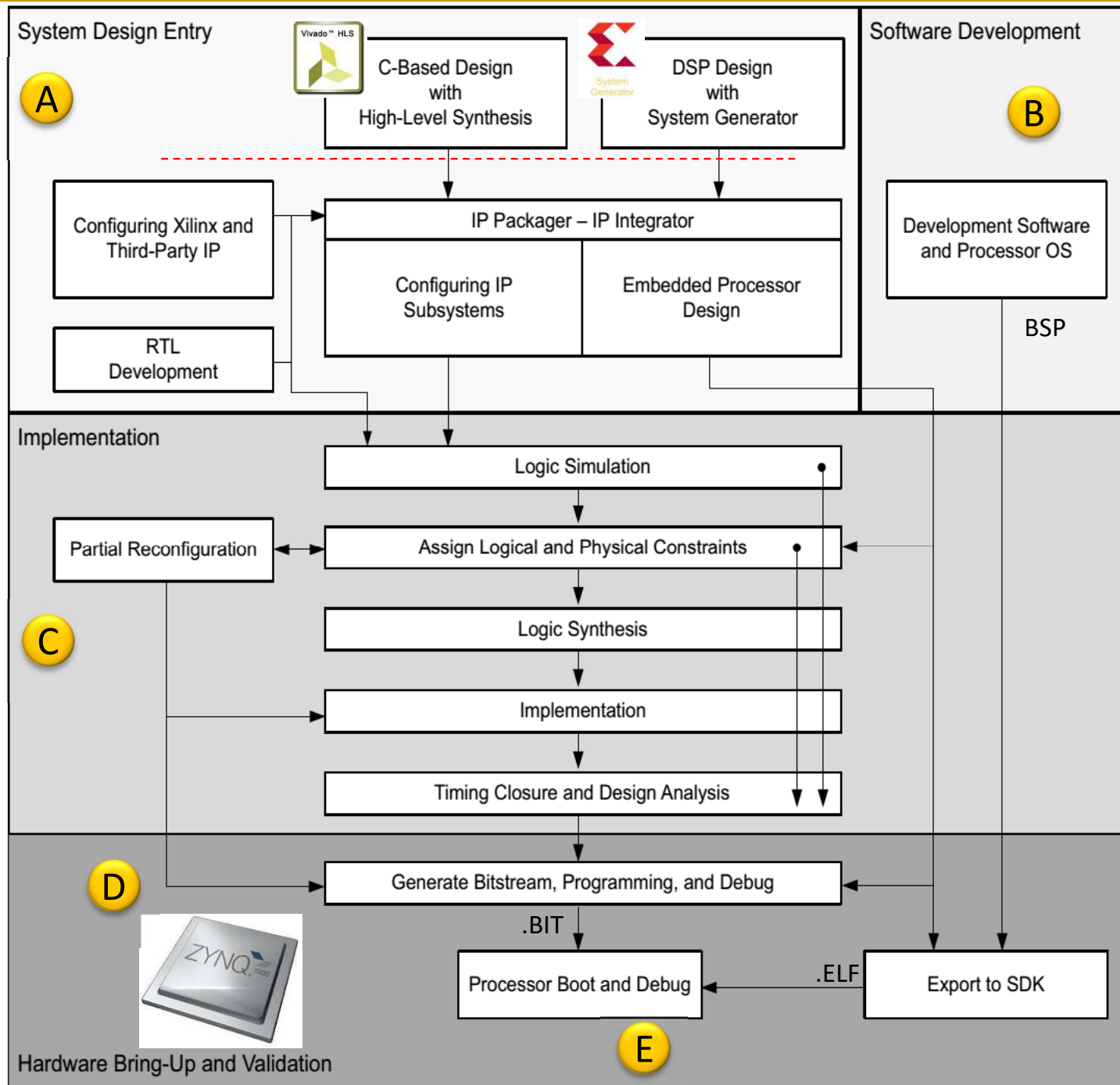
D. Konfigurálás és vizsgálat

- bitstream (.BIT) generálása és konfigurálása FPGA-n

E. Konfiguráció betöltése (opcionális)

- Külső Flash memória inicializálása, ill. boot procedura

Beágyazott rendszer fejlesztés folyamata II.

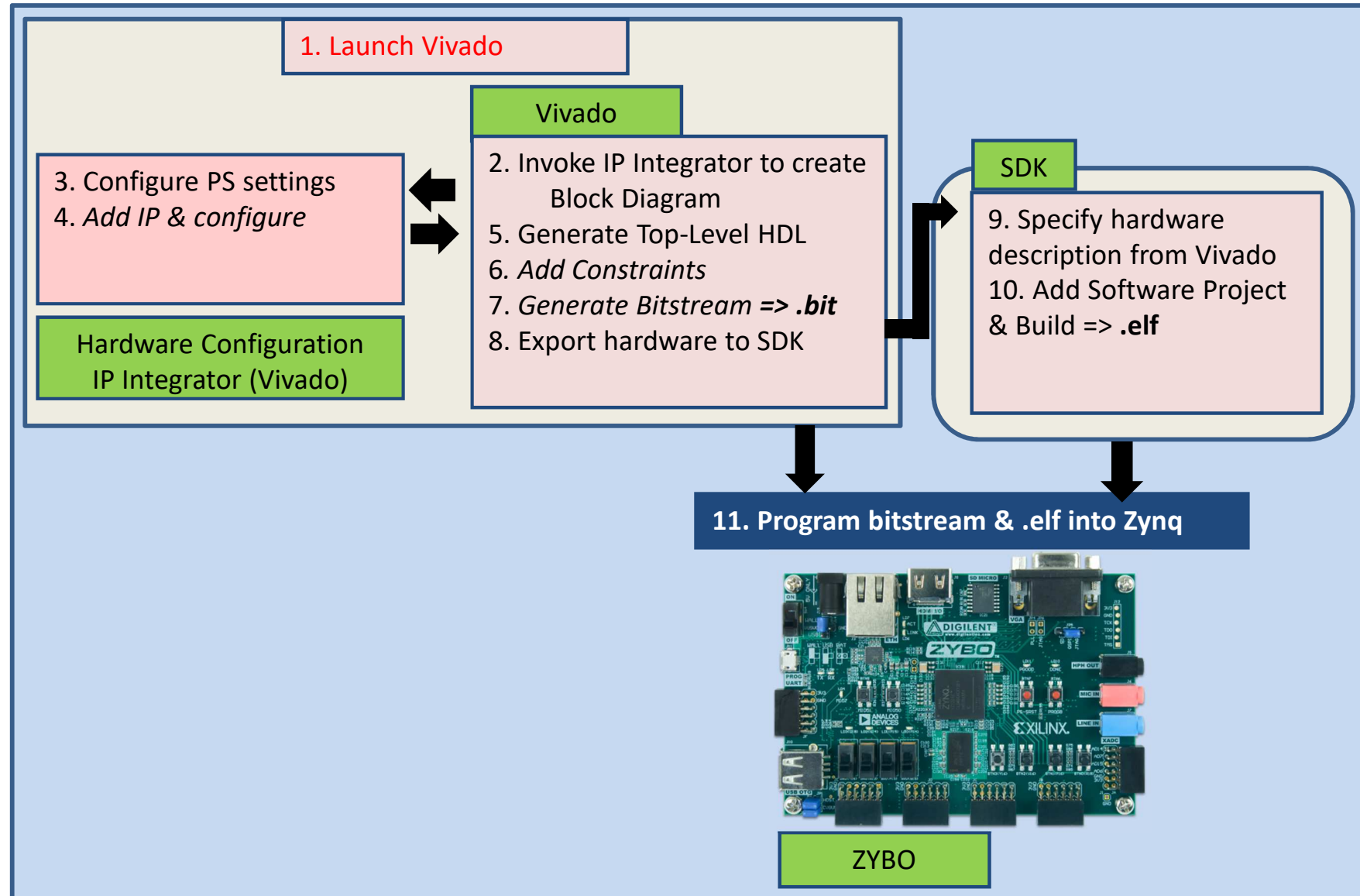


IP:
Intellectual
Property



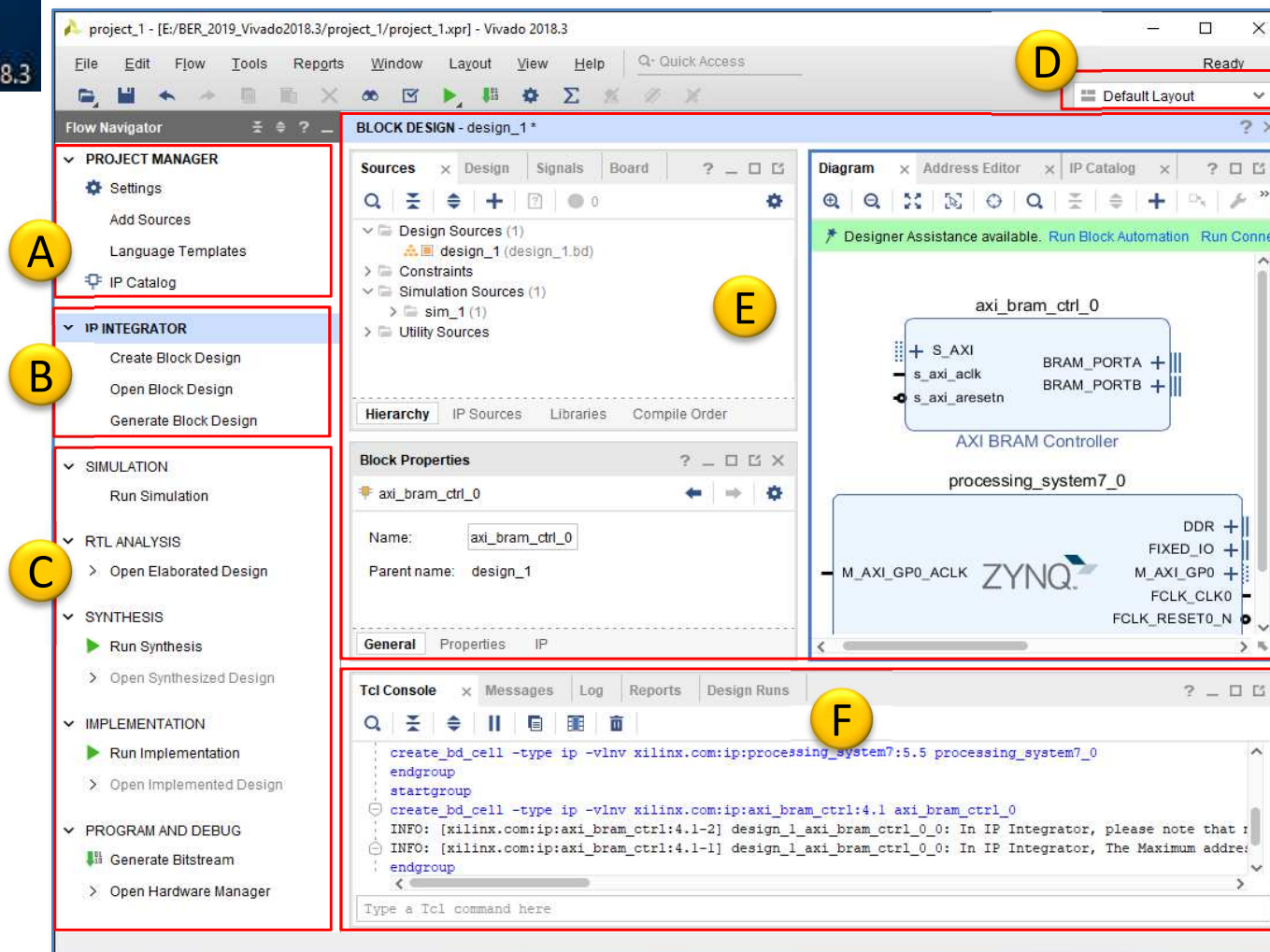
Beágyazott rendszer fejlesztés

Szemléltető példa



Vivado GUI* felülete

- **A:** Project Management
- **B:** IP Integrator
- **C:** FPGA Flow
- **D:** Layout Selection
- **E:** Project view/Preview Panel
- **F:** Console, Messages, Logs



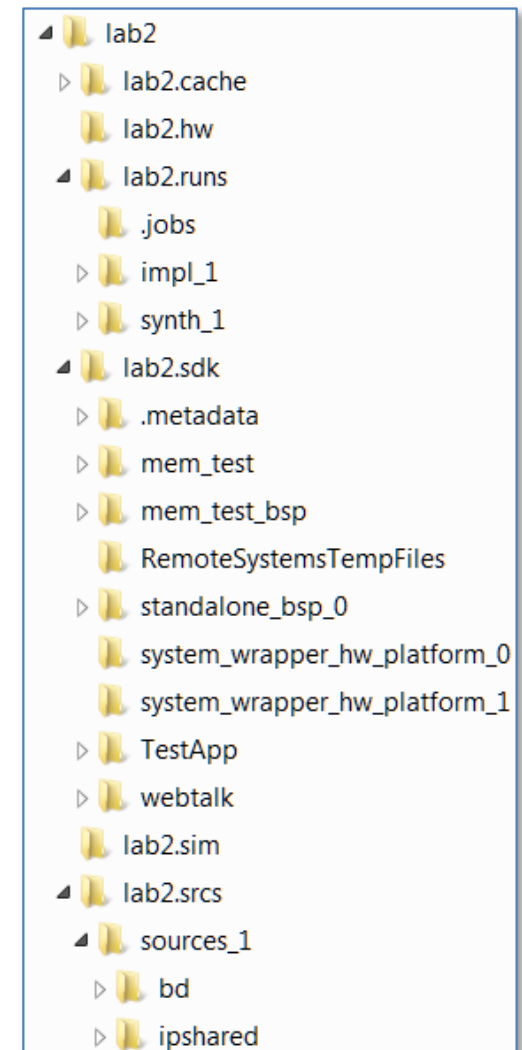
*Megjegyzés: Vivado keretrendszer kétféle módon lehet használni

- **Project mode:** projekt management szemléletmód, GUI-val (ezt használjuk!)
- **Non-project:** script parancsos szemléletmód, GUI nélkül, parancs sorban

Fontosabb Vivado könyvtárak

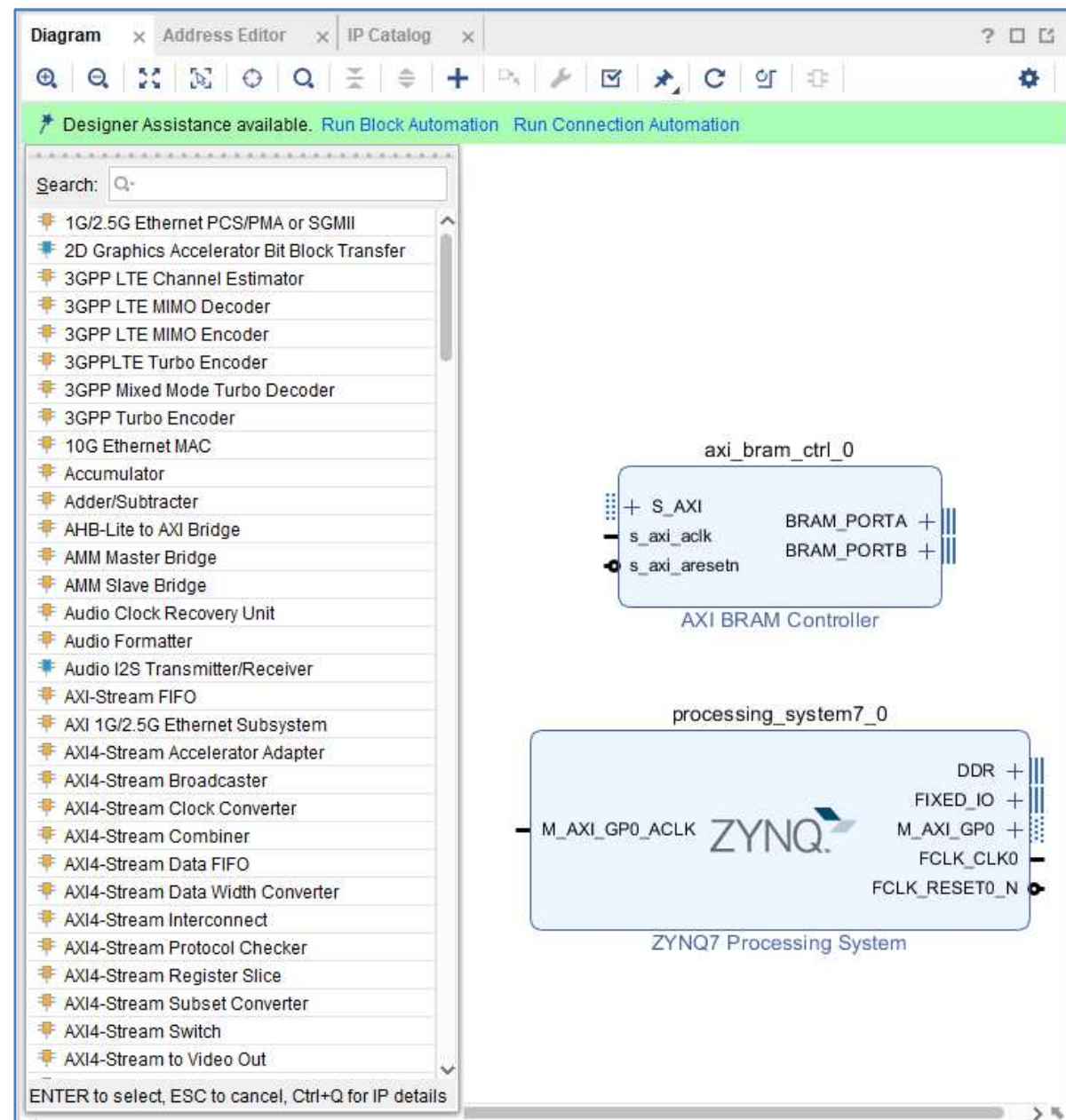


- **Top-level** könyvtárban elérhető
 - **.xpr** = Vivado Project File (xml), log, journal
- **/<név>.srcs**
 - Project source files, IP Integrator fileok
- **/<név>.sim**
 - Szimulációs fileok
- **/<név>.runs**
 - szintézis, implementációs futtatások
- **/<név>.sdk**
 - SDK Export directory, Hardware Platform (xml)
- **/<név>.cache**
 - Ideiglenes fileok (temp)

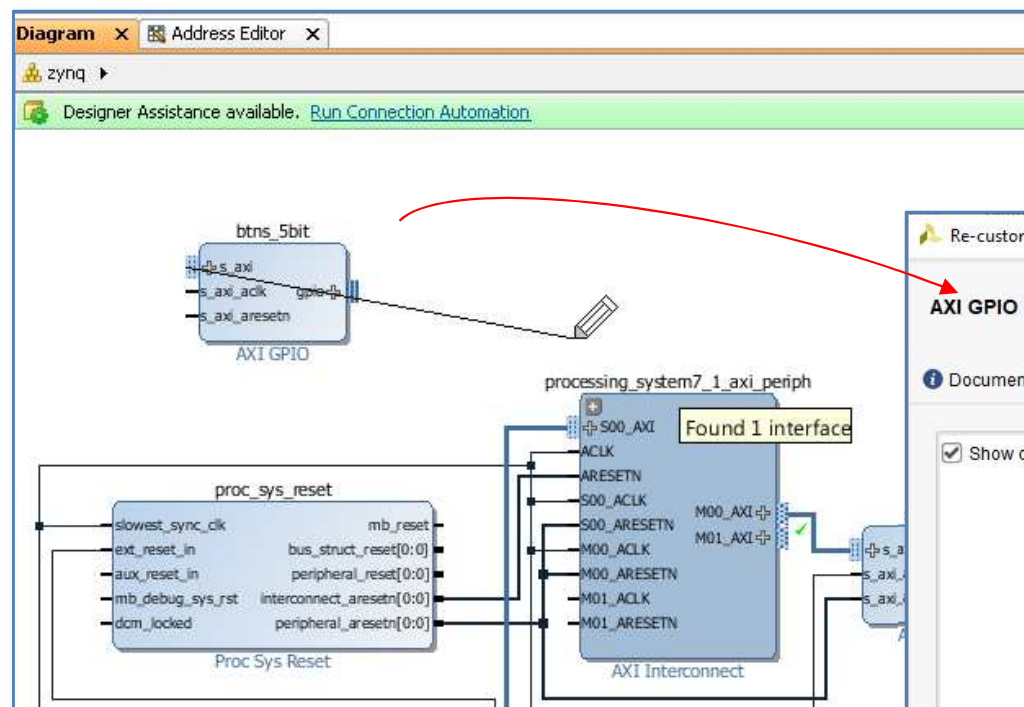


IPI – IP Integrátor

- **IP – Intellectual Property = szellemi termék**
- IP katalógus
 - Drag&Drop
- Hatékony GUI
 - Auto router
 - Redraw
 - Hibás vezetékek jelzése
 - Blokk csoportosítás, hierarchiák kezelése
- IP Packager
 - Saját, vagy külső partner / 3rd party IP-k importálása



IPI – IP Integrátor (folyt)



Re-customize IP

AXI GPIO (2.0)

Documentation IP Location

☒ Show disabled ports

Component Name axi_gpio_0

Board IP Configuration

GPIO

☐ All Inputs

☐ All Outputs

GPIO Width 32 [1 - 32]

Default Output Value 0x00000000 [0x00000000,0xFFFF]

Default Tri State Value 0xFFFFFFFF [0x00000000,0xFFFF]

☐ Enable Dual Channel

GPIO 2

☐ All Inputs

☐ All Outputs

GPIO Width 32 [1 - 32]

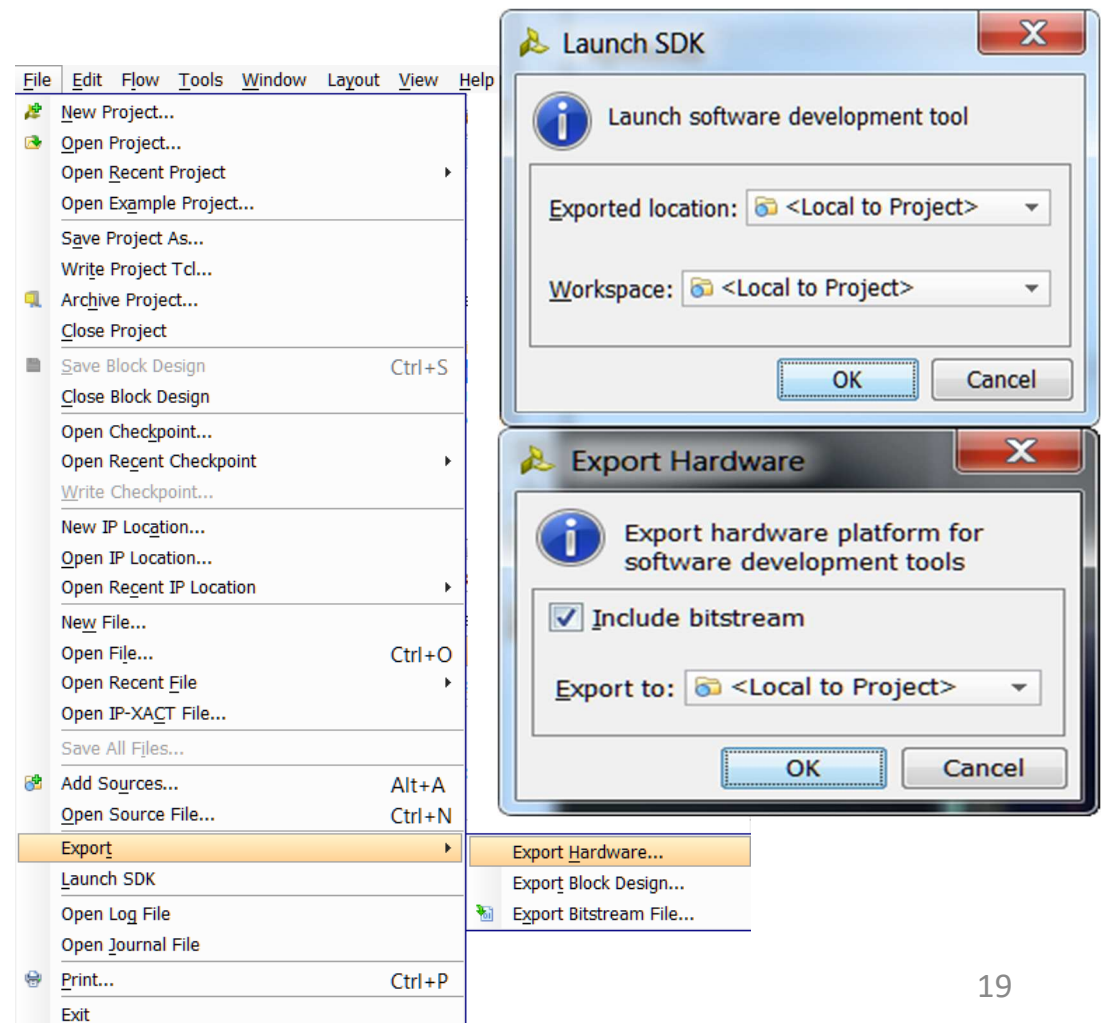
Default Output Value 0x00000000 [0x00000000,0xFFFF]

☐ Enable Interrupt

OK Cancel

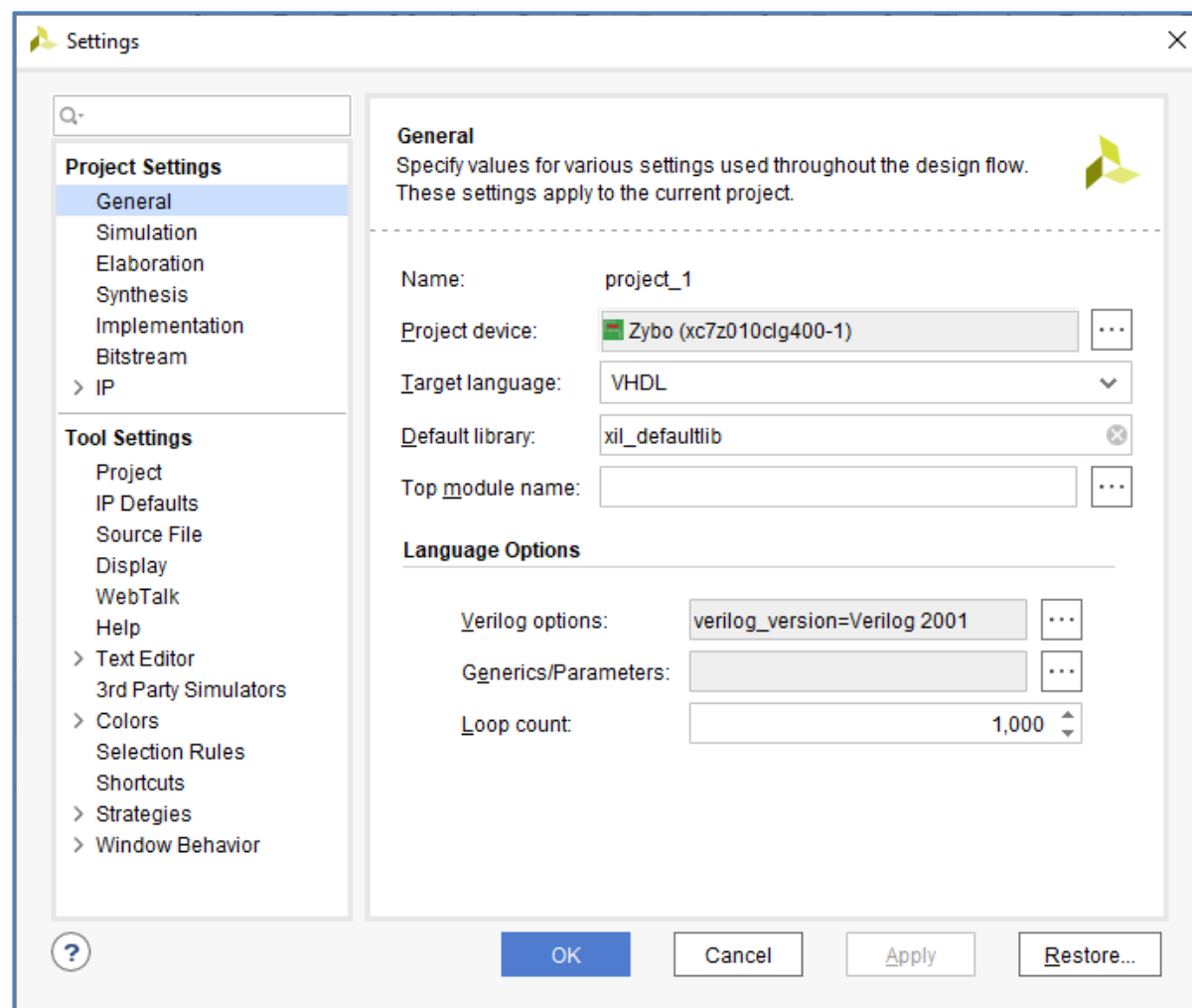
Exporting HW/FW -> SDK

- **.HDF:** hardware definition file:
minden fontos információ a beágyazott rdsz-ről
- **.BIT:** Bitstream is exportálható (opcionális, ha van PL rész)
- **SDK** indítása (de akár külön is)



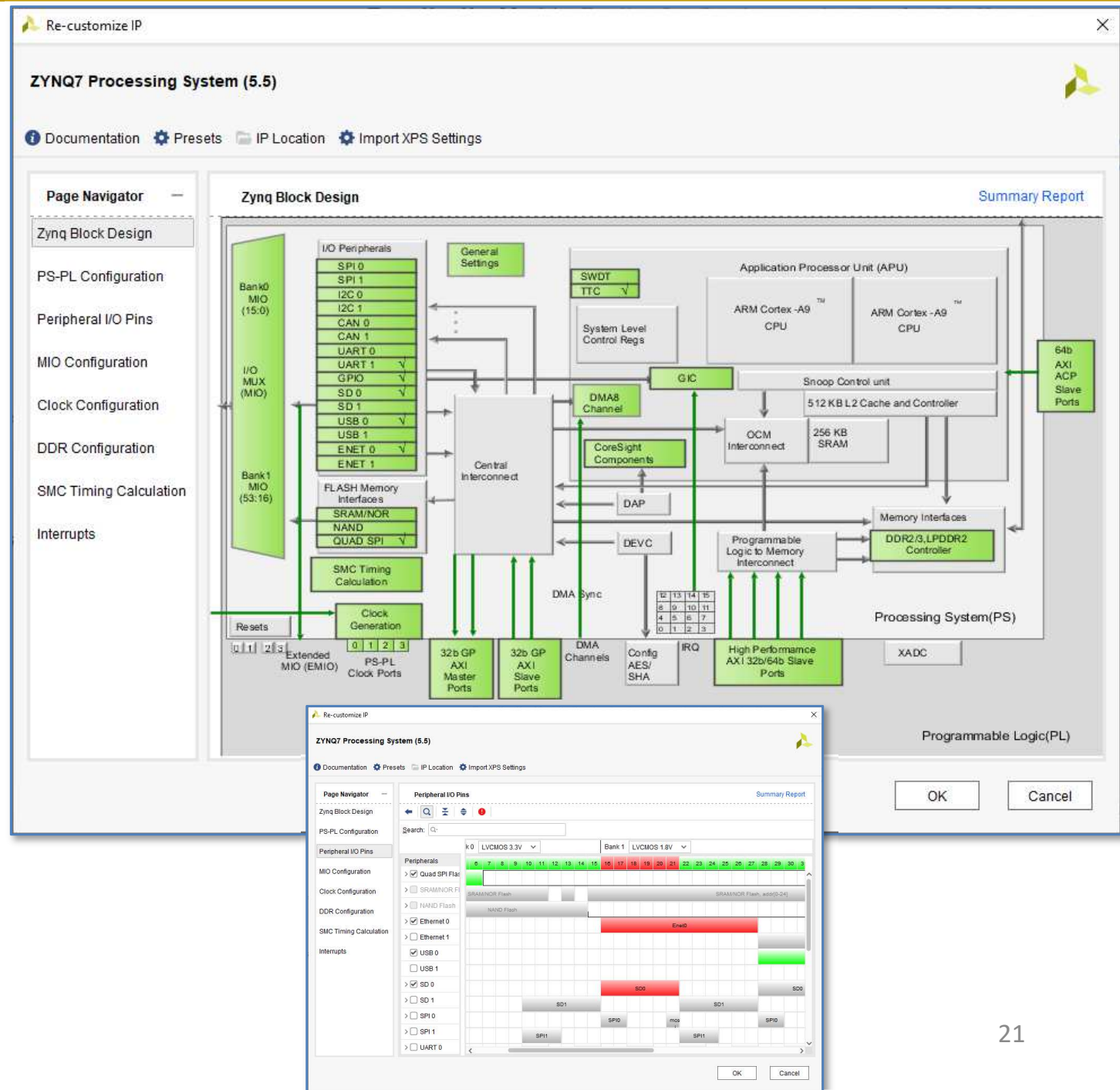
Vivado Projekt beállítások

- Céleszköz beállítása
 - FPGA architektúra,
 - eszköz mérete,
 - tokozás, lábszám
 - Speed grade
- Simulation, Synthesis, Implementation, Bitstream beállítások
- IP repository elérésének beállítása
 - Path-útvonal a külső IP-k megadása
- Tool settings



GUI – Zynq PS nézete

- Zynq PS konfigurációja
 - ARM magok
 - I/O perifériák
 - DDR vezérlő
 - Memória rendszer
- I/O partícionálás:
 - PS, PL részek
 - MIO konfigur
- MemoryMapped konfigurációs regiszterek



Órajel konfiguráció



- Clock Configuration (forrás ARM_PLL, DDR_PLL, IO_PLL)
 - Bemeneti frekvencia állítható (Processzor, DDR)
 - Minden IOP frekvencia állítható
 - PL oldali órajelek engedélyezhetők/ állíthatók
 - Timer(ek) beállítása

The screenshot displays the 'Clock Configuration' tool interface. On the left is a 'Page Navigator' with a tree view containing: Zynq Block Design, PS-PL Configuration, Peripheral I/O Pins, MIO Configuration, **Clock Configuration** (selected), DDR Configuration, SMC Timing Calculation, and Interrupts.

The main area is titled 'Clock Configuration' and has a 'Summary Report' link. It features two tabs: 'Basic Clocking' (active) and 'Advanced Clocking'. Under 'Basic Clocking', the 'Input Frequency (MHz)' is set to 50.000000 and the 'CPU Clock Ratio' is set to 6:2:1.

Below these settings is a table listing various clock components and their configurations:

Component	Clock Source	Requested Frequ...	Actual Frequency(...)	Range(MHz)
▼ Processor/Memory Clocks				
CPU	ARM PLL	650	650.000000	50.0 : 667.0
DDR	DDR PLL	525	525.000000	200.000000 : 534.000...
▼ IO Peripheral Clocks				
SMC	IO PLL	100	10.000000	10.000000 : 100.000000
QSPI	IO PLL	200	200.000000	10.000000 : 200.000000
ENET0	IO PLL	1000 Mbps	125.000000	
ENET1	IO PLL	1000 Mbps	10.000000	
SDIO	IO PLL	50	50.000000	10.000000 : 125.000000
SPI	IO PLL	166.666666	10.000000	0.000000 : 200.000000
> CAN				
▼ PL Fabric Clocks				
☒ FCLK_CLK0	IO PLL	100	100.000000	0.100000 : 250.000000
☐ FCLK_CLK1	IO PLL	50	10.000000	0.100000 : 250.000000

- Vivado beágyazott rendszerfejlesztő eszköz által használt, vagy generált fontosabb fájlok (kiterjesztései):
 - **.XPR** = Xilinx Project File
 - **.HDF** = Hardware Definition File (XML alapú leíró -> SDK bemenete)
 - **.XDC** = Xilinx Constraint File (~ Synopsis .SDC)
 - **.BIT** = Bitstream (configuration file)
 - **.XBD** = Xilinx Board Definition (3rd Party HW leíró fájl) – XML alapú
- Vivado SDK fejlesztő eszköz által generált fontosabb fájlok:
 - **.ELF** = Extensible and Linkable Format
 - **.LD** = Linker Script Download file
 - **.BSP** = Board Support Package (SW + OS)

Általános ismertetés

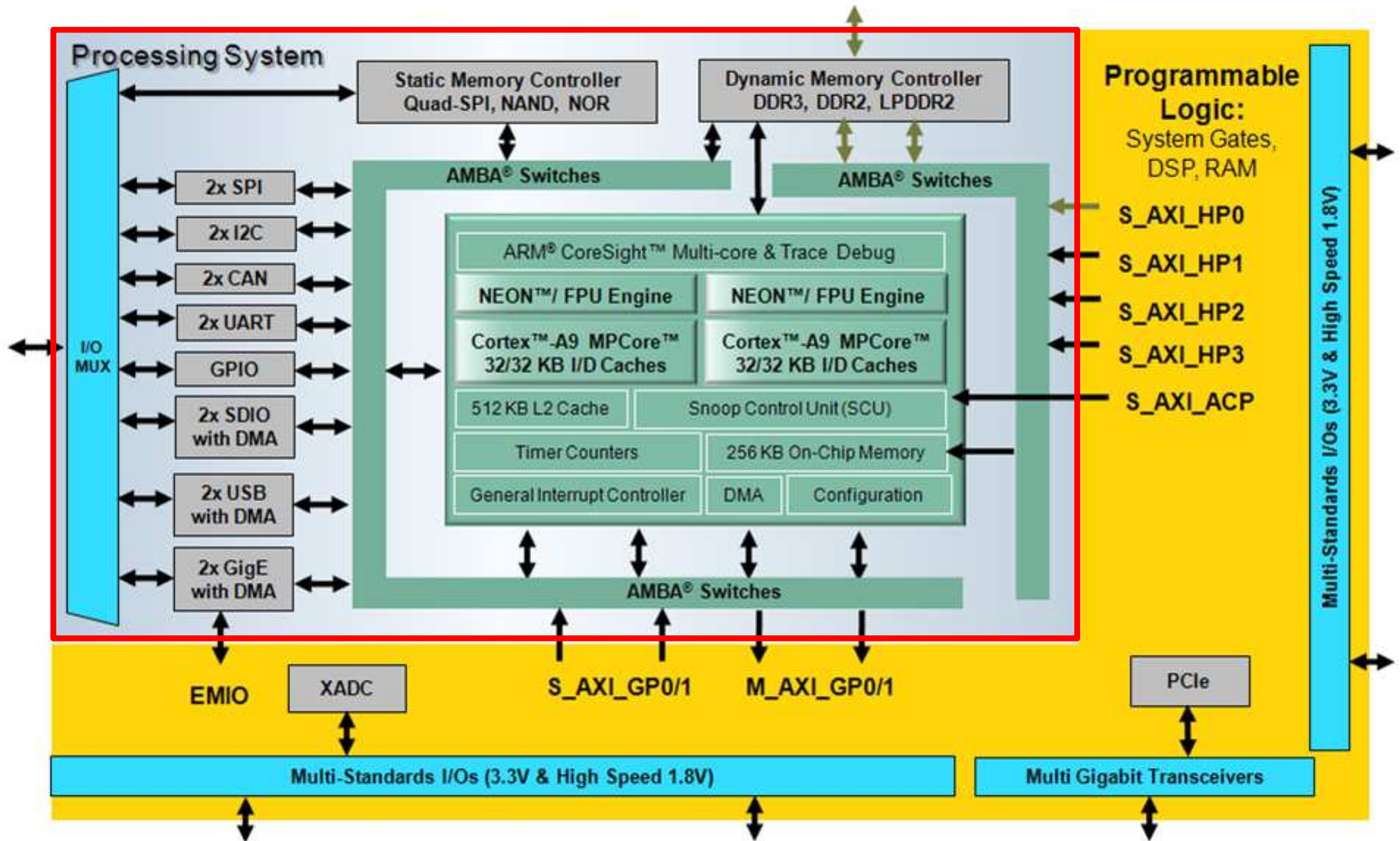
ARM™ BEÁGYAZOTT PROCESSZOR MAG

ARM processzor architektúra



- Zynq PS részén található
- ARM Cortex-A9 processor az **ARMv7-A** alapokon
 - ARMv7 az ARM Instruction Set Architecture (ISA)
 - ARMv7-A: **A**lalmazás Memory Management Unit (MMU)
 - Ezért is hívják még APU-nak, azaz Alkalmazás Processzornak
 - ARMv7-R: **R**real-time Memory Protection Unit (MPU)
 - ARMv7-M: **M**icrocontroller: M0, M3 (2019*)
- ARMv7 ISA utasítások
 - 16 bits/32 bits
 - NEON: ARM's Single Instruction Multiple Data (SIMD) utasítás bővítmény
- ARM Advanced Microcontroller Bus Architecture (AMBA[®]) protocol
 - AXI3: v3 ARM „busz” interfész
 - AXI4: v4 (extended bursts)
- „Cortex” újabb processzor család neve: A9, A53

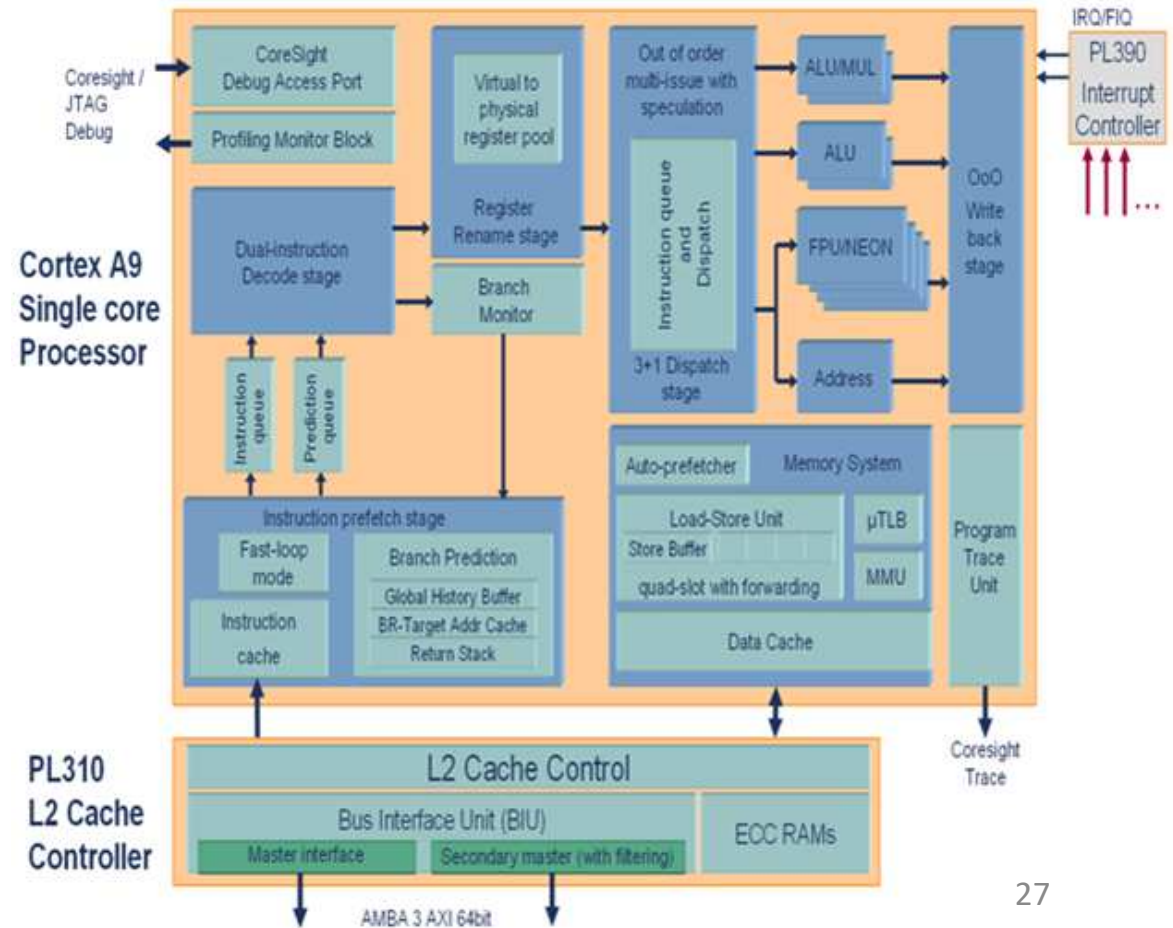
Zynq APSoC blokk diagramm



ARM processzor architektúra



- Zynq PS részén található
- ARM Cortex-A9 processor az **ARMv7-A** alapokon
 - ARMv7 az ARM Instruction Set Architecture (ISA)
 - **ARMv7-A**: Alkalmazás Memory Management Unit (MMU)
 - Ezért is hívják még APU-nak, azaz Alkalmazás Processzornak
 - ARMv7-R: Real time Memory Protection Unit (MPU)
 - ARMv7-M: Microcontroller, skálázható órajellel
- ARMv7 ISA utasításkészlet
 - 16 bits/32 bits
 - NEON: ARM's Single Instruction Multiple Data (SIMD) utasítás bővítmény
- ARM Advanced Microcontroller Bus Architecture (AMBA®) protocol
 - AXI3: v3 ARM „busz” interfész
 - AXI4: v4 (bursts)
- „Cortex” újabb processzor család neve



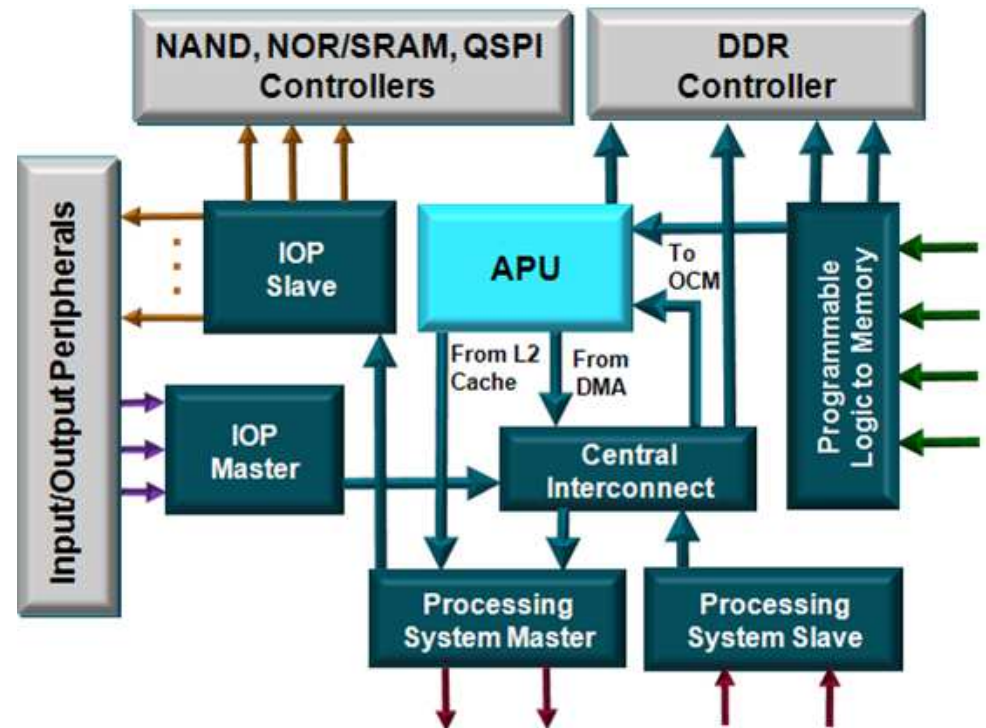
ARM PS/PL memória térképe

- The Cortex-A9 processor uses 32-bit addressing (4Gbyte)
- All **PS** peripherals and PL peripherals are memory mapped to the Cortex-A9 processor cores
- All slave **PL** peripherals will be located between 4000_0000 and 7FFF_FFFF (connected to **GP0**) and 8000_0000 and BFFF_FFFF (connected to **GP1**)

FFFC_0000 to FFFF_FFFF	OCM
FD00_0000 to FFFB_FFFF	Reserved
FC00_0000 to FCFF_FFFF	Quad SPI linear address
F8F0_3000 to FBFF_FFFF	Reserved
F890_0000 to F8F0_2FFF	CPU Private registers
F801_0000 to F88F_FFFF	Reserved
F800_1000 to F880_FFFF	PS System registers,
F800_0C00 to F800_0FFF	Reserved
F800_0000 to F800_0BFF	SLCR Registers
E600_0000 to F7FF_FFFF	Reserved
E100_0000 to E5FF_FFFF	SMC Memory
E030_0000 to E0FF_FFFF	Reserved
E000_0000 to E02F_FFFF	IO Peripherals
C000_0000 to DFFF_FFFF	Reserved
8000_0000 to BFFF_FFFF	PL (MAXI_GP1)
4000_0000 to 7FFF_FFFF	PL (MAXI_GP0)
0010_0000 to 3FFF_FFFF	DDR(address not filtered by SCU)
0004_0000 to 000F_FFFF	DDR(address filtered by SCU)
0000_0000 to 0003_FFFF	OCM

PS fontosabb részei

- Application processing unit (APU)
- I/O peripherals (IOP)
 - Multiplexed I/O (MIO), extended multiplexed I/O (EMIO). Max 54 láb!
- Memory interfaces
- PS interconnect
- DMA
- Timers
 - Public and private
- General interrupt controller (GIC)
- On-chip memory (OCM): RAM
- Debug controller: CoreSight



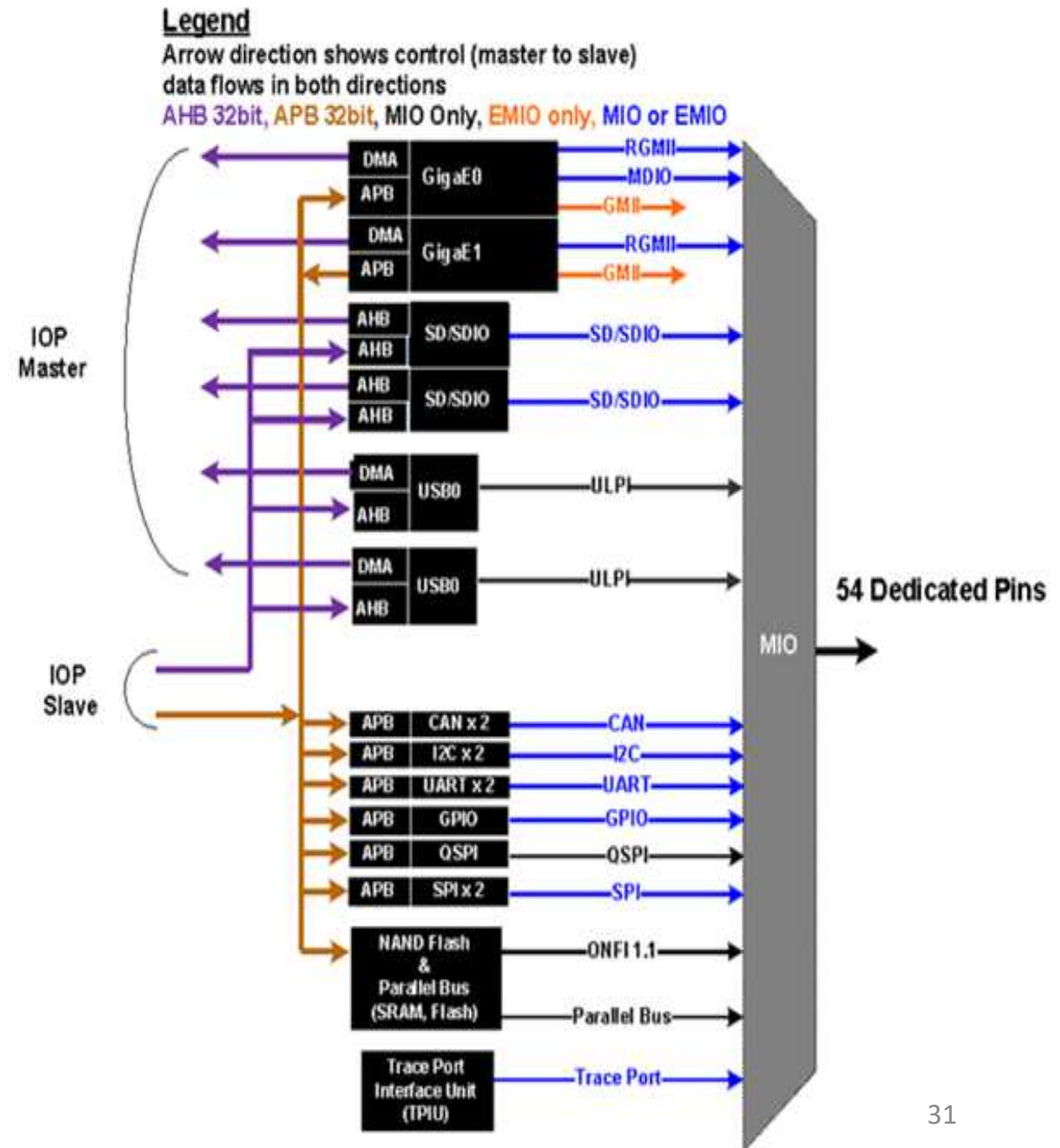
Boot sorrend



- CPU0 boots from OCM ROM; CPU1 goes into a sleep state
- On-chip boot loader in OCM ROM (Stage 0 boot)
- Processor loads **First Stage Boot Loader (FSBL)** from external flash memory:
 - NOR
 - NAND
 - Quad-SPI
 - SD Card
 - JTAG; not a memory device—used for debug only
 - (Boot source selected via package bootstrapping pins)
- Optional secure boot mode allows the loading of encrypted software from the flash boot memory

IOP – IO perifériák

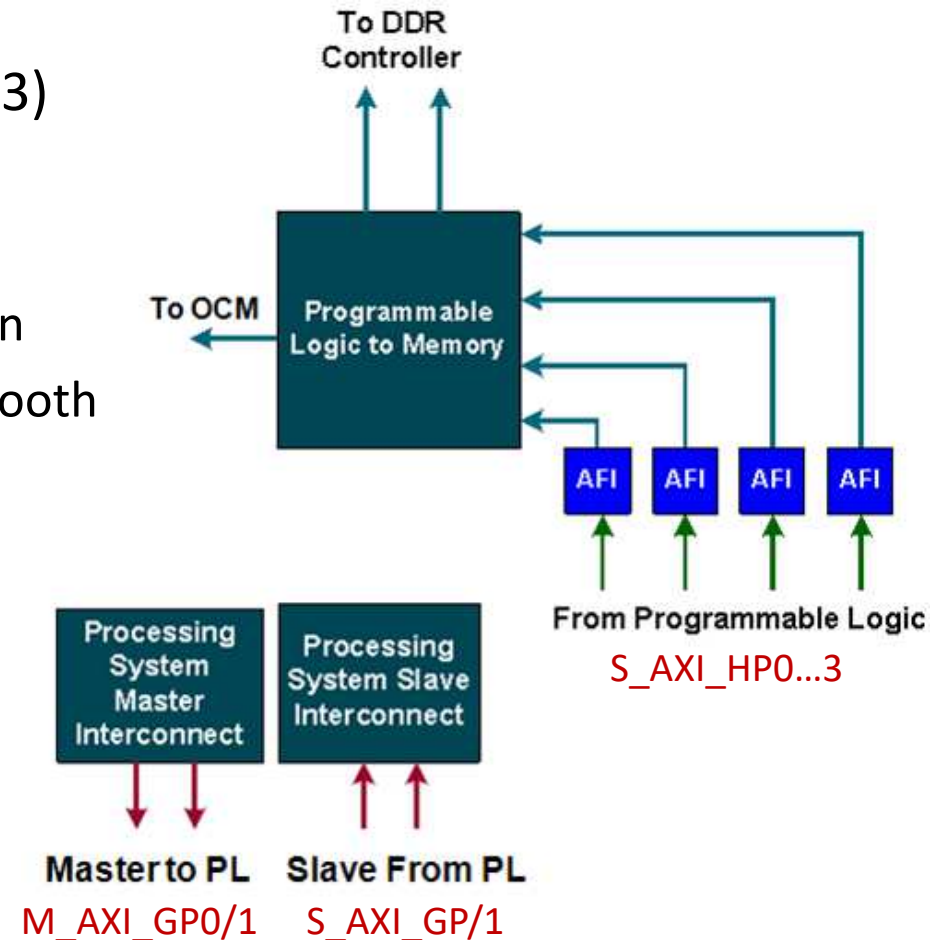
- 2 GigE
- 2 USB
- 2 SPI
- 2 SD/SDIO
- 2 CAN
- 2 I2C
- 2 UART
- 4 db 32-bit GPIOs
- Static memories
 - NAND, NOR/SRAM,
 - Quad SPI
- Trace ports



PS-PL összeköttetések (hidak)



- **AXI high-performance slave portok (HP0-HP3)**
 - 4 Configurable 32-bit or 64-bit data width
 - Access to OCM (1) and DDR (2) only
 - Conversion to processing system clock domain
 - AXI FIFO Interface (AFI) are FIFOs (1KB) to smooth large data transfers
- **AXI general-purpose portok (GP0-GP1)**
 - 2masters from PS to PL
 - 2slaves from PL to PS
 - 32-bit data width
 - Conversation and sync to processing system clock domain

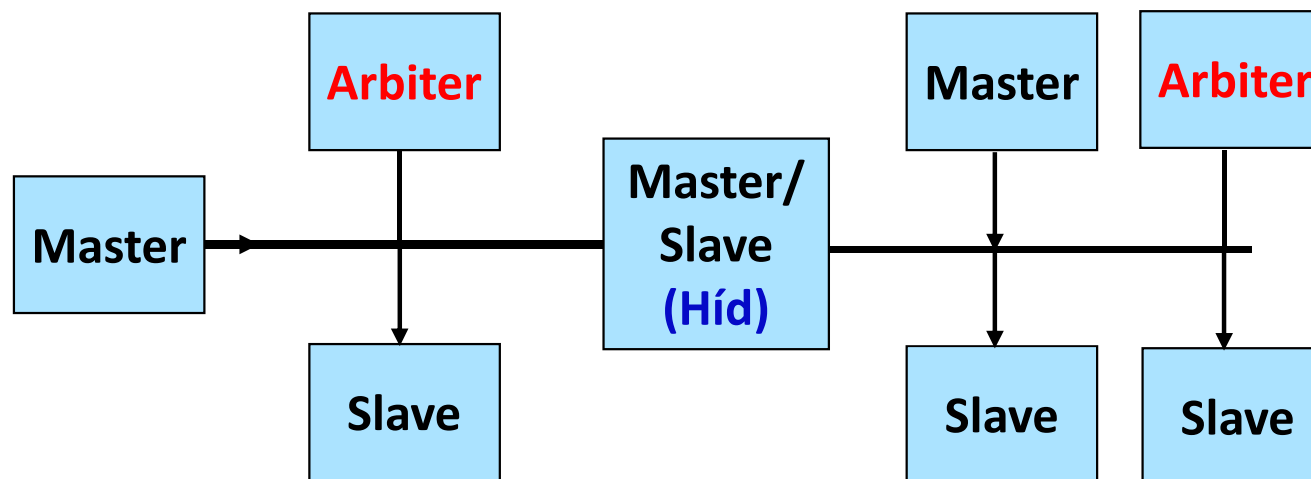


Általános ismertetés

PROCESSZOR MAGOKHOZ CSATLAKOZÓ BUSZRENDSZEREK

Buszok „1x1”

- Busz: egy olyan több-vezetékes útvonal, amely a következő információkat hordozza:
 - cím, adat, utasítás, vezérlési információ
- Processzor (magok) és az IP perifériák összekapcsolására
- Perifériák osztályozása a „kezdeményezés” alapján:
 - Arbiter (A) – arbitrációs egység, master (M) - kezdeményező, vagy slave (S) - fogadó, vagy mindkettő master/slave is egyben (híd=bridge)



Buszok „1x1”

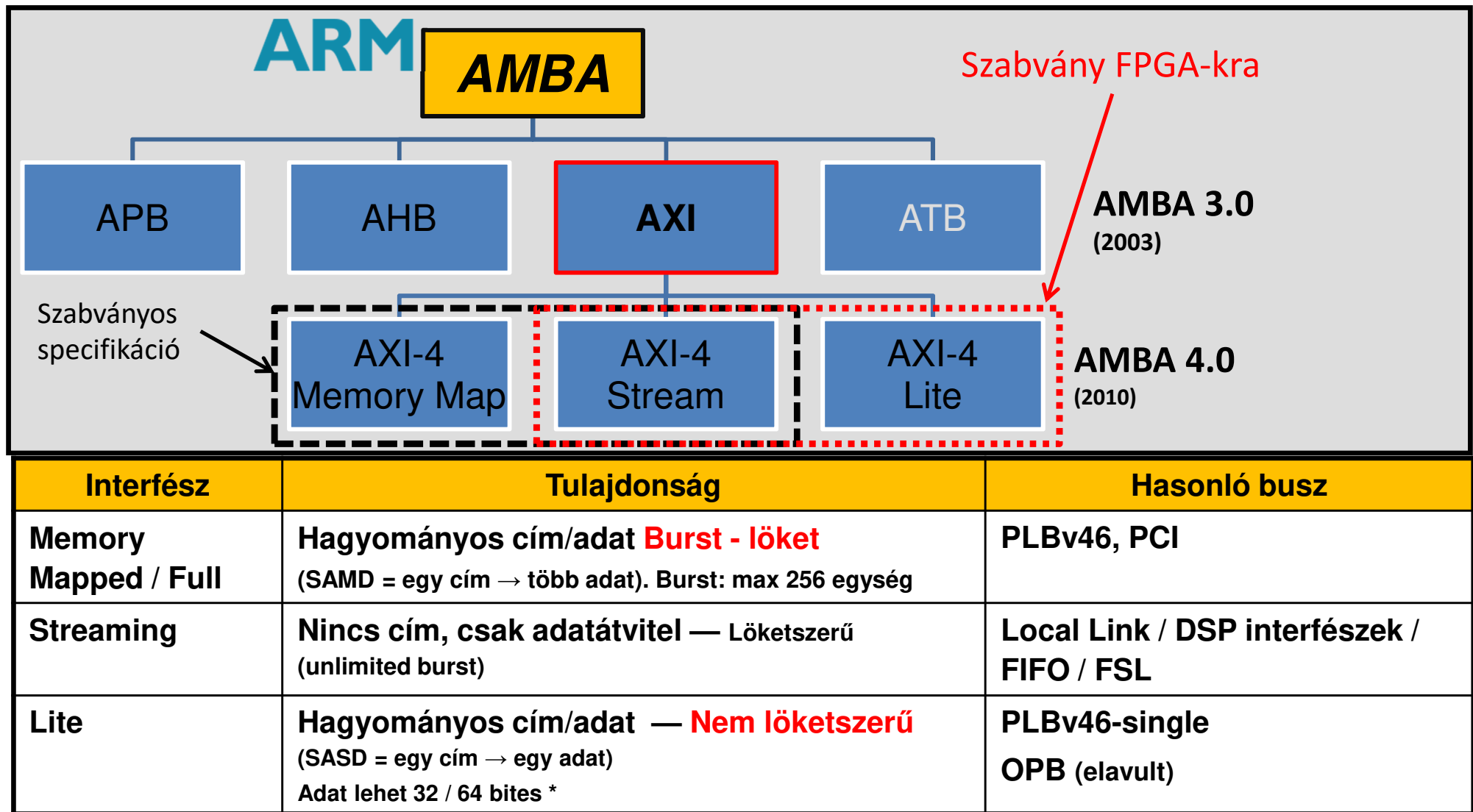


- Busz master (M) : képesség busz tranzakció indítására
- Busz slave (S): csak válaszol egy kérésre
- Busz arbitráció (A): egy 3-lépéses döntési folyamat
 - A master (M) eszköz a BR (bus request) vonalon keresztül benyújtja busz foglalási igényét az arbiter (A) felé a soron következő tranzakcióhoz
 - Az arbiter (A) folyamatosan monitorozza (poll) az igényeket, és kiad egy BG (bus grant) jelet annak a master egységnek, amelynek pl. legnagyobb volt a fix-prioritása (ha több is benyújtotta egyszerre igényét)
 - Az busz-igénylő master (M) egység veszi a BG jelet, majd pedig akkor indítja el a **tranzakciót** (írás-olvasás) a slave (S) egység felé, amikor az előző master (M) egység befejezi az aktuális tranzakcióját
- Arbitráció: döntési mechanizmus = Ki legyen a master a következő tranzakciónál? Döntés alapja lehet:
 - Fix-prioritású,
 - round-robin, vagy akár
 - hibrid módszer

ARM AMBA – AXI interfész



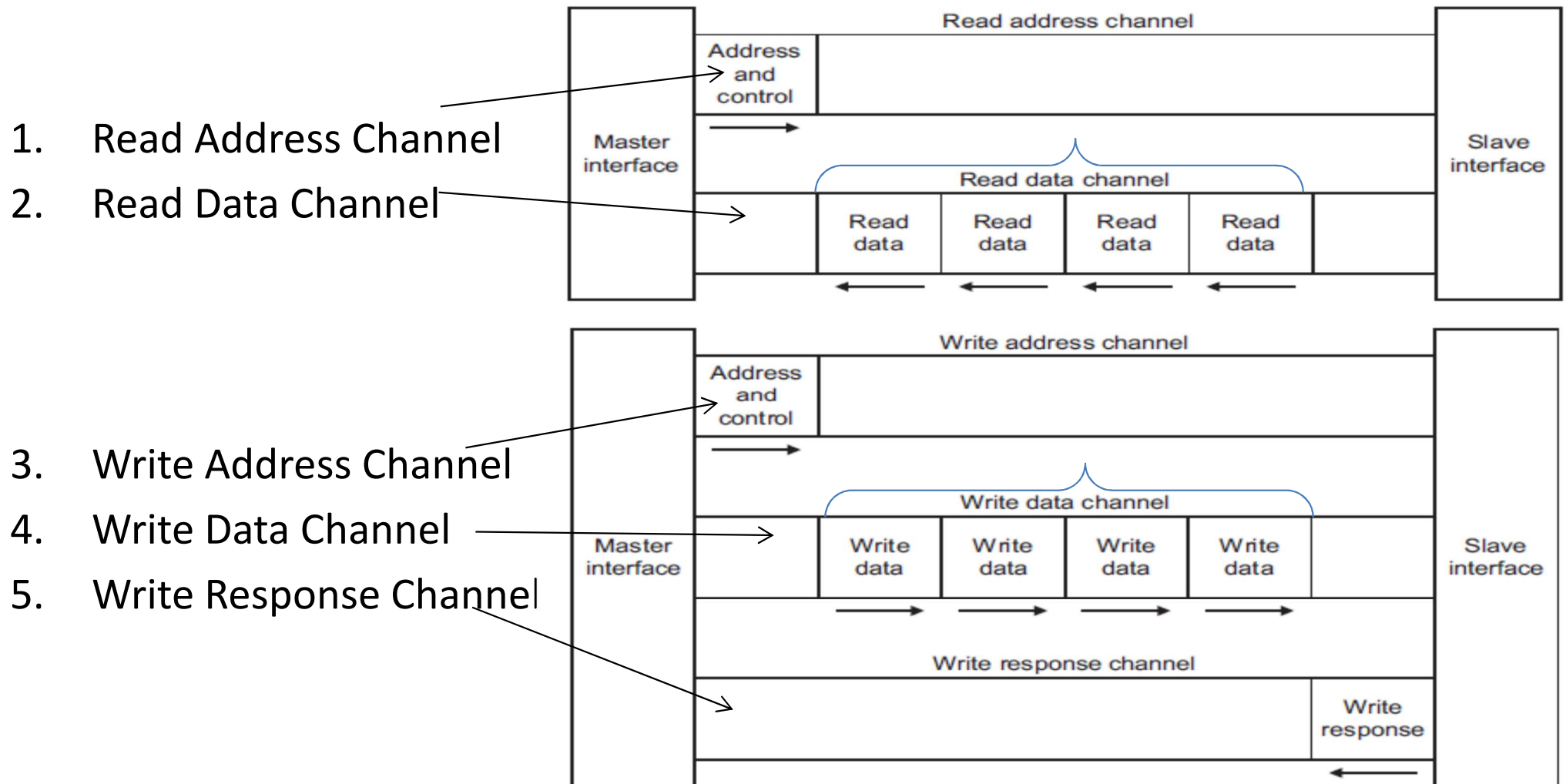
AMBA: Advanced Microcontroller Bus Architecture



*Xilinx IP 32-bites adatvitel támogatása

- Szabványos **interfész** és **protokoll** definíció egyben,
- Széles körben használt,
- Nem tekinthető busz összeköttetésnek!
- Az AXI specifikációk inkább **interfészként** definiálják (és nem pedig busz összeköttetésként!) :
 - Nem azt adja meg, hogyan kell összekötni a rendszerben lévő IP perifériákat, hanem
 - Egy IP periféria be-, ki-meneteit definiálja.

AXI4 alap mód (memory mapped)

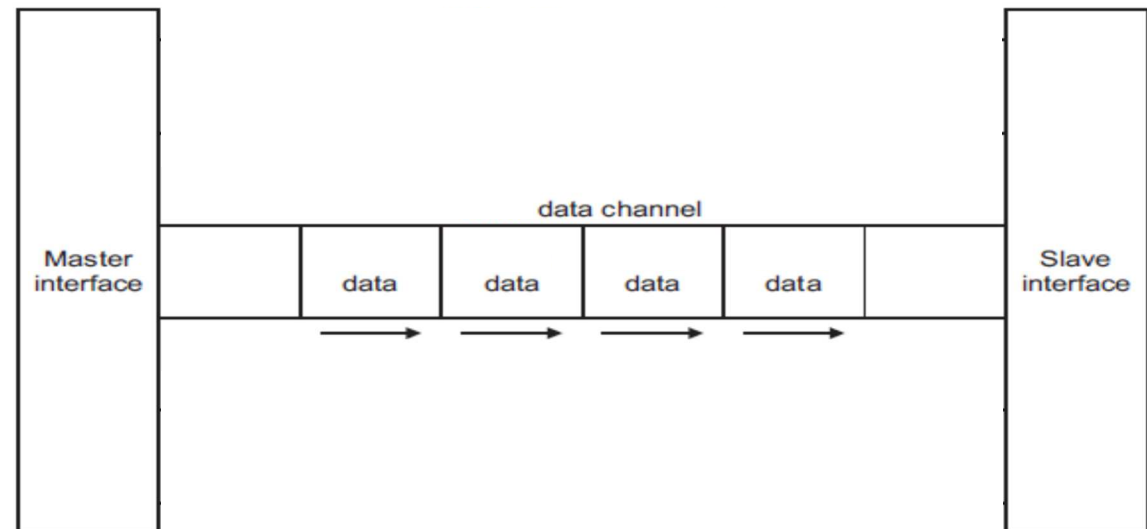


AXI4 Stream



- **No address channel**, no read and write, always just master to slave
 - Effectively an AXI4 “write data” channel
- Unlimited burst length
 - AXI4 max 256
 - AXI4-Lite does not burst
- Virtually same signaling as AXI Data Channels
 - Protocol allows merging, packing, width conversion
 - Supports sparse, continuous, aligned, unaligned streams

AXI4-Stream Transfer

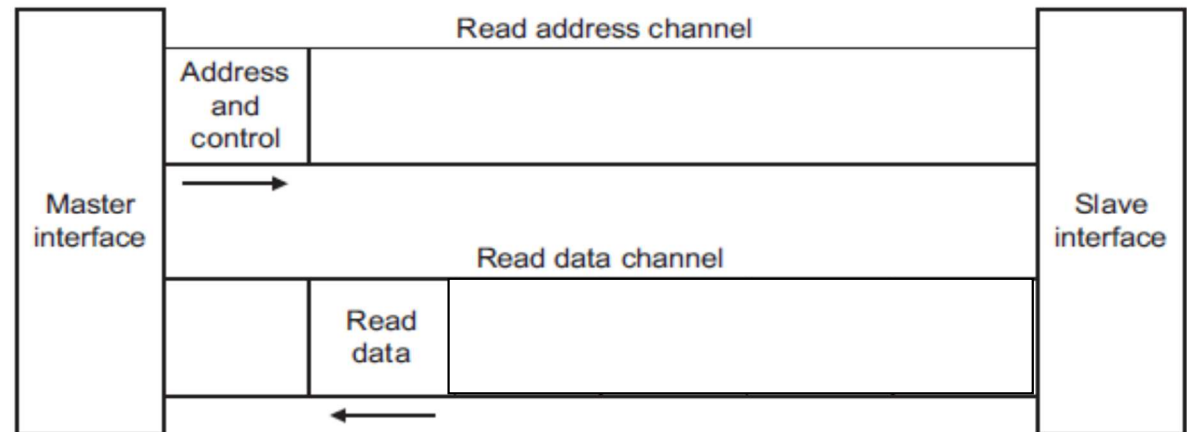


AXI4 Lite mód

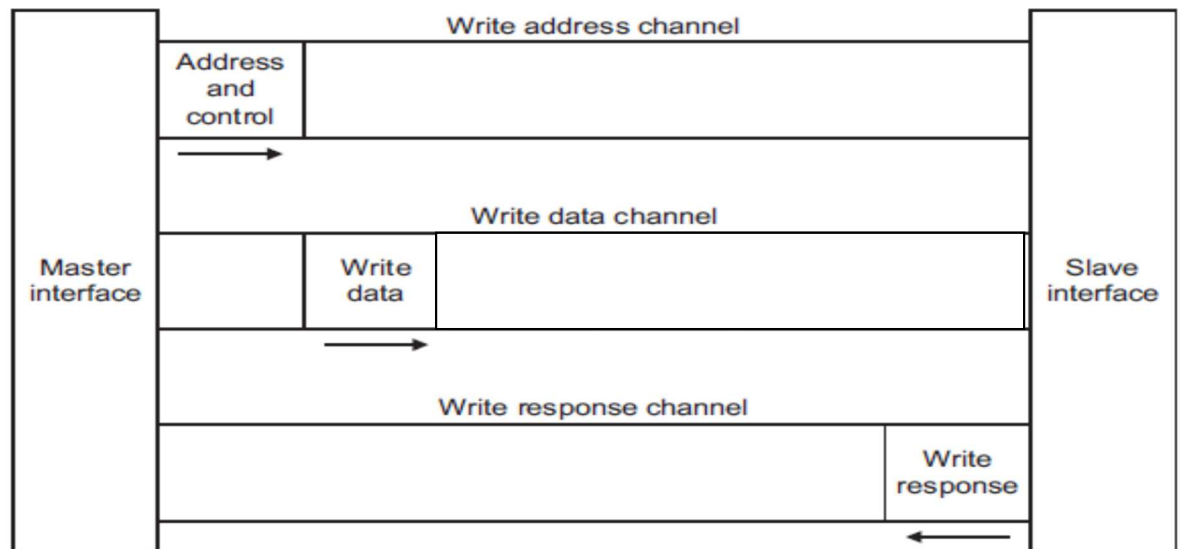


- No burst
 - Ex. pass control information
- Data width 32 or 64 only
 - Xilinx IP only supports 32-bits
- Very small footprint
- Bridging to AXI4 handled automatically by AXI_Interconnect (if needed)

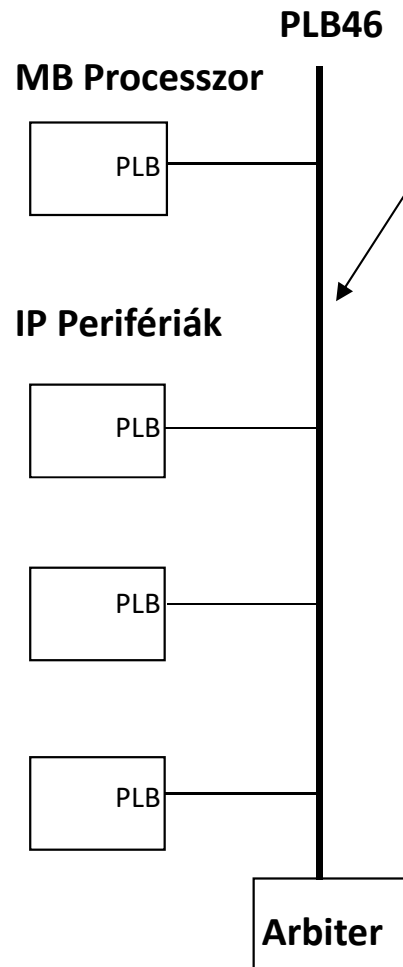
AXI4-Lite Read



AXI4-Lite Write



AXI vs. PLB

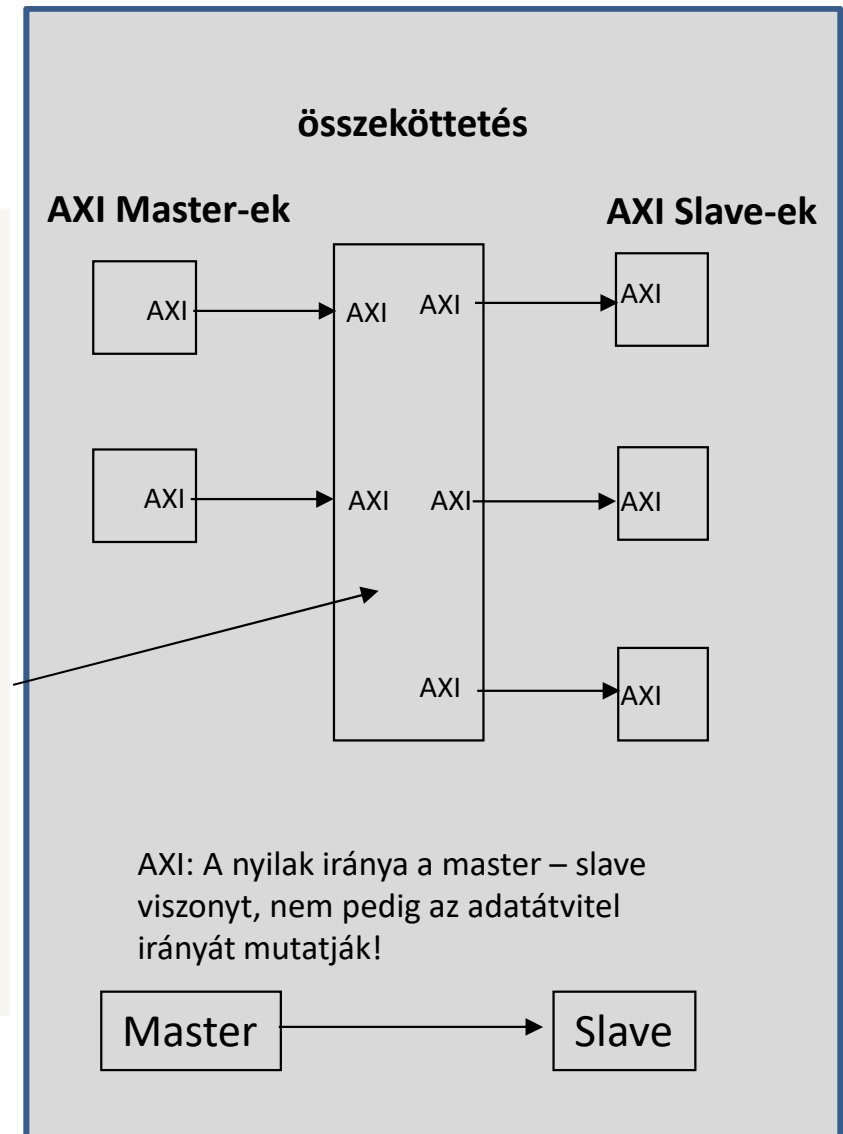


Osztott elérésű busz

AXI Interconnect IP

- Implementáció nincsen egzakt módon leírva
- Minden cég saját “AXI interconnect IP”-t hozhat létre
- Xilinx is megépítette a saját AXI rendszerét

AXI = interfész specifikáció (és nem busz specifikáció, mint a PLB!)



- Az LMB **1 órajel ciklus alatt végbemenő adat/utasítás elérést** biztosít a MicroBlaze processzornak
 - on-chip, két-port-os memóriák (BRAM-ok)
 - saját dedikált adat/utasítás busszal
- Egyszerű szinkron protokoll
 - DLMB: Adat interfész LMB (csak BRAM)
 - ILMB: Utasítás interfész LMB (csak BRAM)

- **MicroBlaze/ARM** az AXI-interfész támogatott a Vivado-ban
- **AXI**: interfész nagy sebességű pont-pont összeköttetést biztosít
- **LMB**: processzor lokális memóriája (adat, verem, és megszakítás kezelő rutinok számára). Fix, és determinisztikus késleltetésű.
- MicroBlaze esetén:
 - **AXI**: Little és Big-endian is lehet

Rövid ismertetés

BEÁGYAZOTT SZOFTVER-FEJLESZTŐ KÖRNYEZET (XILINX VIVADO SDK)

Software Development Kit (SDK)



Általános tulajdonságai (röviden):

- Teljes-értékű integrált SW fejlesztő környezet
 - Eclipse-alapú nyílt forráskódú IDE
- Az Vivado Project Navigatortól különálló Xilinx modul
 - Csak beágyazott SW alkalmazások fejlesztésére használható
- Beágyazott HW/FW rendszerek debug-olását biztosítja
- Kifinomult SW környezet, mely egyszerre támogat:
 - Több-processzor magos platformokat,
 - Több BSP-t használó SW rendszereket,
 - Több SW alkalmazást futtató rendszereket.
- Hatékony C/C++ forráskód szerkesztő (editor) és hiba-kereső (navigátor)
- * SDK használata részletesen a BER_04. fóliától



- Fordító (builder) modul:
 - Forrás fájlok fordítása és összerendelése (linker script: **.LD**)
 - Alapértelmezett fordítási beállítások (option) generálódnak egy beágyazott SW alkalmazás létrehozásakor: Debug, Release vagy Profile konfigurációk kiválasztása
 - Később ezek a beállítások szabadon változtathatók, finom-hangolhatók (custom build)
 - Fordítási típusok: Standard Make, Managed Make (**makefile.**)
- Futtató modul:
 - Meghatározható: alkalmazás futtatása (Run /+ Profile), vagy Debug
 - Céleszköz beállításai (Target Connection settings)
 - Futtatható file: **.ELF + Executable Load File**
- Hibakereső (debug) modul:
 - Hibakereső (gdb) indítása, alkalmazás betöltése, majd pedig hibakeresés
 - Hibakeresés állapotáról információkat ad (Debug nézet)
- Kereső (search) modul:
 - Segíti az alkalmazás fejlesztését
- Segítő modul:
 - Online-help

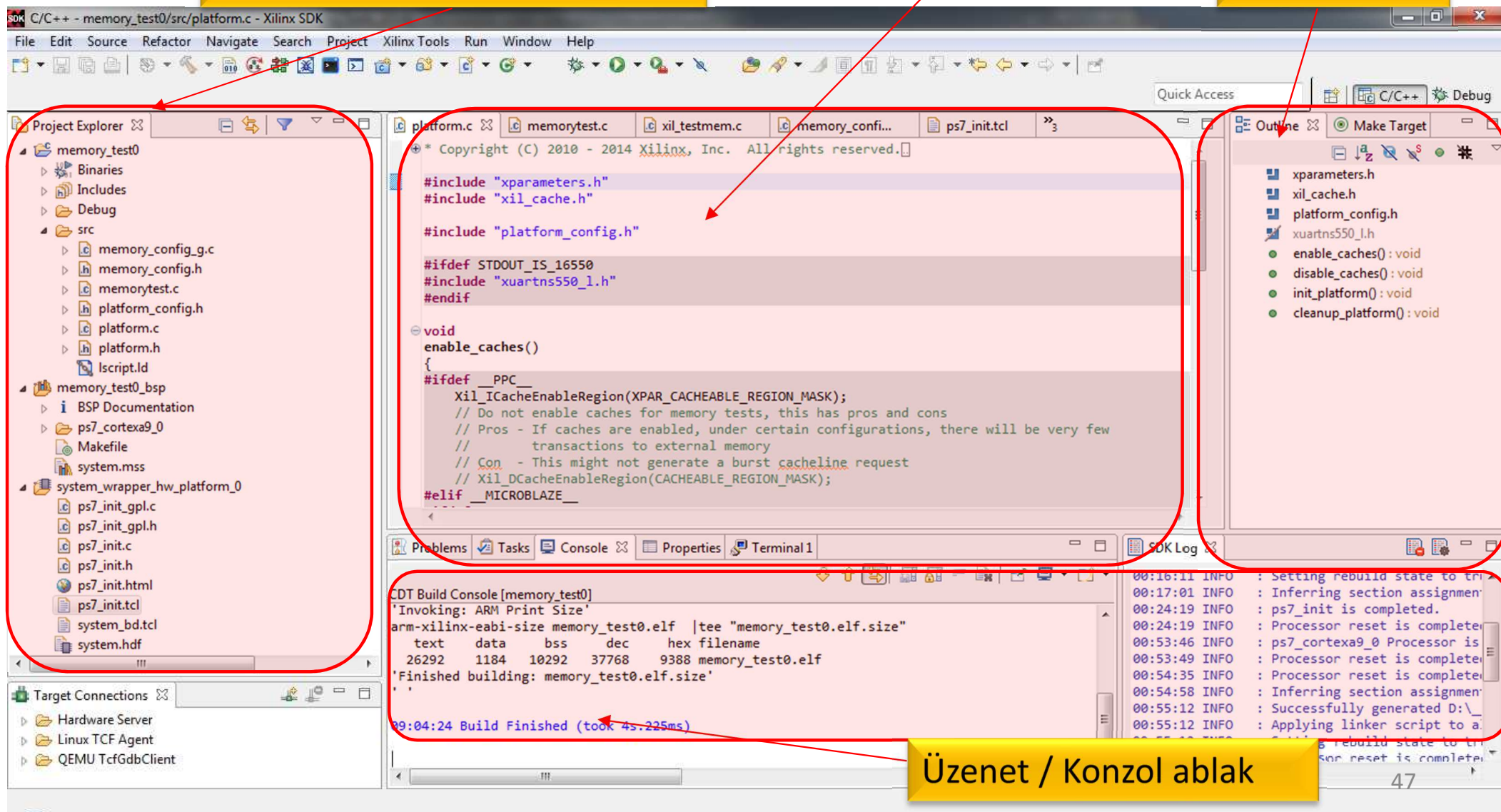
Vivado SDK GUI

Project Explorer (3 szint):

- HW Platform
- BSP: Board Support Package
- Alkalmazás projekt

Forráskód ablak

Debug ablak



The screenshot displays the Vivado SDK GUI with the following components:

- Project Explorer (Left):** Shows a hierarchical view of the project structure. The 'memory_test0' project is expanded, showing sub-projects like 'memory_test0_bsp' and 'system_wrapper_hw_platform_0'. The 'src' directory contains files like 'memory_config.g.c', 'memory_config.h', 'memorytest.c', 'platform_config.h', 'platform.c', 'platform.h', and 'Iscript.ld'.
- Source Editor (Center):** Displays the source code for 'platform.c'. The code includes headers like 'xparameters.h', 'xil_cache.h', and 'platform_config.h'. It defines a function 'enable_caches()' which uses 'Xil_ICacheEnableRegion' and 'Xil_DCacheEnableRegion' to enable caches for different processor architectures (PPC and MICROBLAZE).
- Debug Console (Bottom):** Shows the output of the build process. It includes messages like 'Invoking: ARM Print Size', 'arm-xilinx-eabi-size memory_test0.elf', and 'Finished building: memory_test0.elf.size'. The console also shows the size of the generated ELF file: 'text data bss dec hex filename 26292 1184 10292 37768 9388 memory_test0.elf'.
- Outline (Right):** Provides a summary of the code structure, listing variables and functions such as 'enable_caches()', 'disable_caches()', 'init_platform()', and 'cleanup_platform()'.

Üzenet / Konzol ablak

Vivado SDK – SW management

3 fő terület:

1. **Compiler/Linker**
beállítások -
alkalmazásra
2. **BSP: Board**
Support Package
– SW platformra
3. **Linker Script**
generálás
(memória
kiosztások)

