EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

FPGA-BASED EMBEDDED SYSTEM DEVELOPMENT

(VEMIVIB334BR)

Created by Zsolt Voroshazi, PhD

voroshazi.zsolt@mik.uni-pannon.hu

Updated: 13. Feb. 2024.

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Strukturális és Beruházási Alapok

BEFEKTETÉS A JÖVŐBE

VIVADO HLx Editions

2. XILINX VIVADO-VITIS

Embedded System development environment

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Strukturális és Beruházási Alapok

BEFEKTETÉS A JÖVŐBE

Topics covered

1. Introduction – Embedded Systems
2. FPGAs, Digilent ZyBo development platform
3. **Embedded System - Firmware development environment (Xilinx Vivado – „EDK" Embedded Development)**
4. **Embedded System - Software development environment (Xilinx VITIS – „SDK")**
5. Embedded Base System Build (and Board Bring-Up)
6. Adding Peripherals (from IP database) to BSB
7. Adding Custom (=own) Peripherals to BSB
8. Development, testing and debugging of software applications – Xilinx VITIS (SDK)
9. Design and Development of Complex IP cores and applications (e.g. camera/video/ audio controllers)
10. HW-SW co-simulation and testing(Xilinx Vivado ChipScope)
11. Embedded Operation System I.: Application development, testing, device drivers, and booting
12. Embedded Operation System II.: setting and starting Linux system on ARM/MicroBlaze processor

Requirements

- Digital Circuits and Computer Architectures
- Design Methods: Learn how to use the Xilinx Vivado Design Suite (HLx) development environment
- Basic C / C ++ knowledge
  - Programming I. / II.
- Basics of Digital Systems and Computer Architectures
- Basics of HDL knowledge (e.g. VHDL language)
  - Design Methods with Programmable Logic Devices (BSc in Informatics, BSc in Electrical Engineering)

Xilinx product portfolio (2020)

- Previously: ISE and XPS Platform Studio (till 2013)
  - Support up to FPGA Series-6 and older devices. Obsoleted!
  - Switching to newer Vivado/VITIS 202x.y is recommended!

- **Vivado Design Suite HLx** (support APSoCs, UltraScale MPSoCs and FPGA Series-7
  - Latest version: 2023.2
  - We use Vivado 2020.2 in the laboratory work
- Vivado HLS: High Level Synthesis
  - Support C/C++, OpenCV languages

- ***VITIS** = Unified Software Platform 2020.2
- SDx: Software Defined
  - SDAccel, SDSoC (reVision), SDNet (fully integrated into the new VITIS environment!)
    - Support Cloud, C/C++, OpenCL

Xilinx Vivado + VITIS

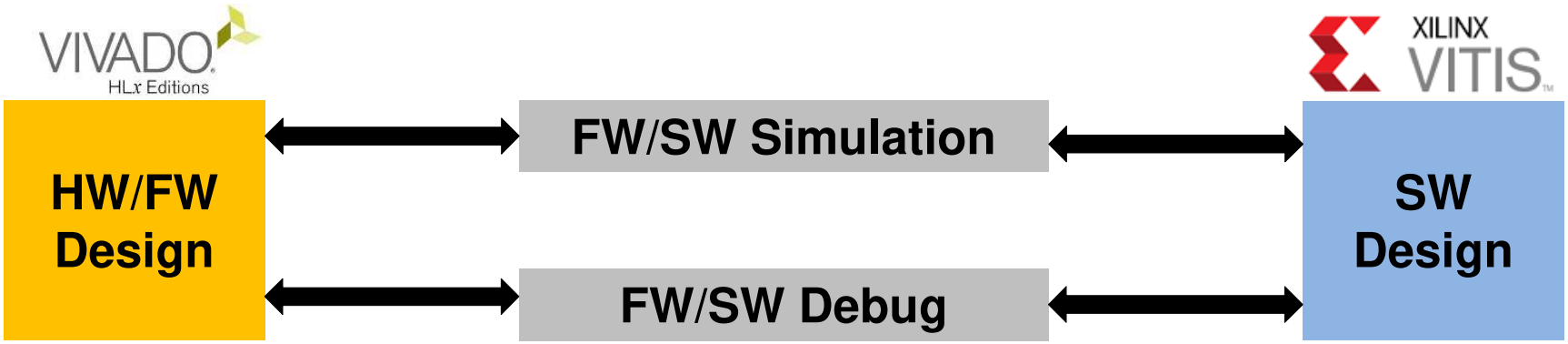
We use Vivado Design Suite HLx 2020.2 in the laboratory work.

- Vivado Project Navigator: framework - IDE, various modules can be invoked (aspects of project management as PlanAhead)
  - **Embedded designer / IPI = IP integrator (~EDK) – designing and developing Embedded firmware ecosystem (FPGA logic + ARM cores)**
    - Xilinx IP Catalogue (free, or purchased IPs)
    - Custom-own, or 3rd Party IPs
  - Timing Analyzer
  - FPGA Editor
  - **HW Manager– Download Bitstream (FPGA configuration)**
  - **ILA: Integrated Logic Analyzer (~oscilloscope)**
- **VITIS as Unified Software Platform – Embedded software development (~SDK)**
  - Eclipse based IDE environment

Xilinx Vivado + VITIS

Two main modules in the IDE framework:

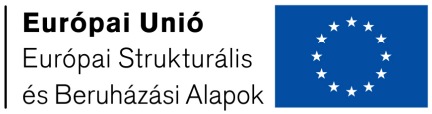
- **Vivado Embedded Designer / IP Integrator** (~ EDK): Xilinx FPGA-based integrated development environment for assembling and parameterizing embedded systems (at **HW / FW** level) for rapid prototype development
- **VITIS** as **SDK**: Eclipse-based integrated **SW** development environment to support embedded processors, and accelerators (e.g. MicroBlaze™, ARM™):
  - SW: C / C ++ applications,
  - GCC, GPP (built-in/external) compilers,
  - GNU debugger
  - Support for embedded OS (Linux distributions).



XILINX VIVADO AS EMBEDDED SYSTEM DEVELOPMENT

General description – FPGA design flow

MAGYARORSZÁG KORMÁNYA



BEFEKTETÉS A JÖVŐBE

References

📖 Xilinx Vivado Design Suite development tool:

- https://www.xilinx.com/products/design-tools/vivado.html#documentation

📖 Vivado Getting Started Guide (UG-910)

- https://docs.xilinx.com/v/u/2020.1-English/ug910-vivado-getting-started

📖 Zynq-7000 All Programmable SoC Software Developers Guide (UG-821)

- https://docs.xilinx.com/r/en-US/ug821-zynq-7000-swdev

📖 Zynq-7000 All Programmable SoC Embedded Design tutorial (UG-1165)

- https://docs.xilinx.com/v/u/2020.1-English/ug1165-zynq-embedded-design-tutorial

📖 Zynq-7000 APSoC Concepts, Tools, and Techniques (Hands-on Guide)

- https://docs.xilinx.com/v/u/en-US/ug873-zynq-ctt

Xilinx Vivado

Aims:

- Development of embedded systems (firmware) for Xilinx FPGAs (eg Spartan-7, Virtex-7, Kintex-7) or APSoCs (e.g. Zynq-7000 family),
- Description and integration of components (IP), parameterization,
- Methods and steps of HW / FW design implementation,
- Demonstration and learning of development tools,

- VITIS: Introduce SDK as SW development environment, hardware debugging.

Elaboration of Embedded System (FW) in Vivado

- Integrate processor system(s) in FPGA
  - Xilinx MicroBlaze: soft-core architectures, or
  - ARM: hard-core processors
- Design and instantiate IP blocks
- Connect Reset / Clock / Debug ports
- Set peripheral inteconnections
  - AMBA AXI v.4 busz interfész (supported from Vivado 2012.x and EDK 12.x)
- Map addresses in the address space
- Validate layout (=block diagram)

Elaboration of Embedded Software (SW/OS) in VITIS

- Generate SW libraries and drivers
  - C / C ++ language support
  - Compiling a SW application
  - SW drivers (low-high level), routines (interrupt), pre-defined API functions, SW stacks
- Operating system (OS or RTOS)
  - Stand-alone (non-OS) vs. Bare-metal, resp.
  - Embedded Linux OS (PetaLinux), vs. VxWorks RTOS distros, etc.
- Booting procedure

Embedded Systems - Development Flow

A. Embedded HW/FW development (VIVADO)
- IP Integrator: rapid prototype development for target hardware
- Extension of an existing embedded HW-FW system
  - Integrating peripherals in the IP catalog, or using custom (self-made) IP peripherals,
- Generating HDL netlist (description)

B. Implementation
- Synthesis, MAP, PAR, and timing analysis (user constraints)

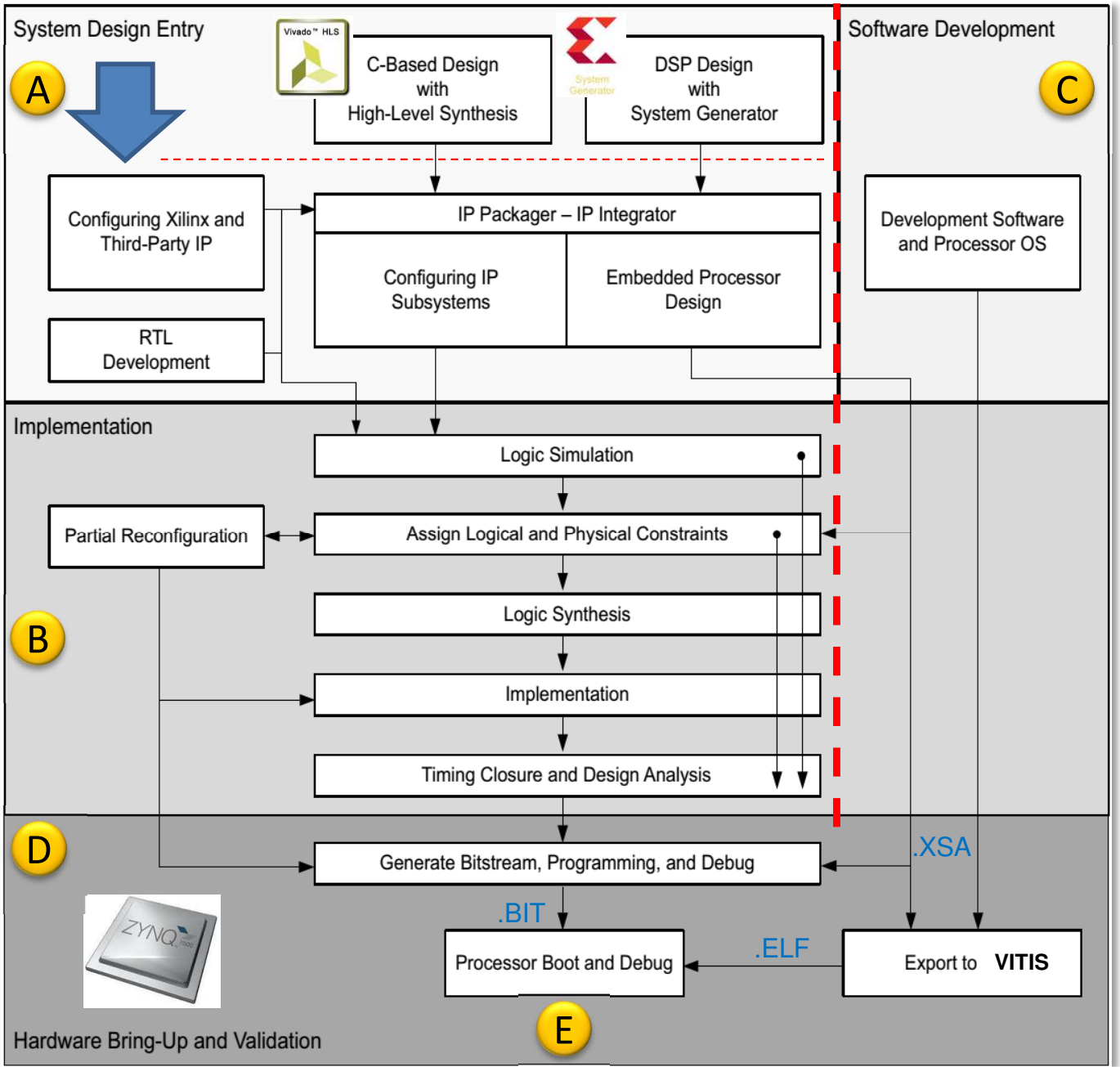
C. Development of embedded SW in SDK (VITIS)
- BSP: Board Support Package / Domain
- Generating device libraries and drivers
- Creating SW Application and Debugging
- Optional: debug application using Xilinx Microprocessor Debug (XMD) and GNU debugger (gdb) on ARM / MicroBlaze cores

D. Configuration and testing
- bitstream (.BIT) generation and configuration on FPGA

E. Load configuration (optional)
- Initialization of external Flash memory, resp. boot procedure

Embedded Systems - Development Flow

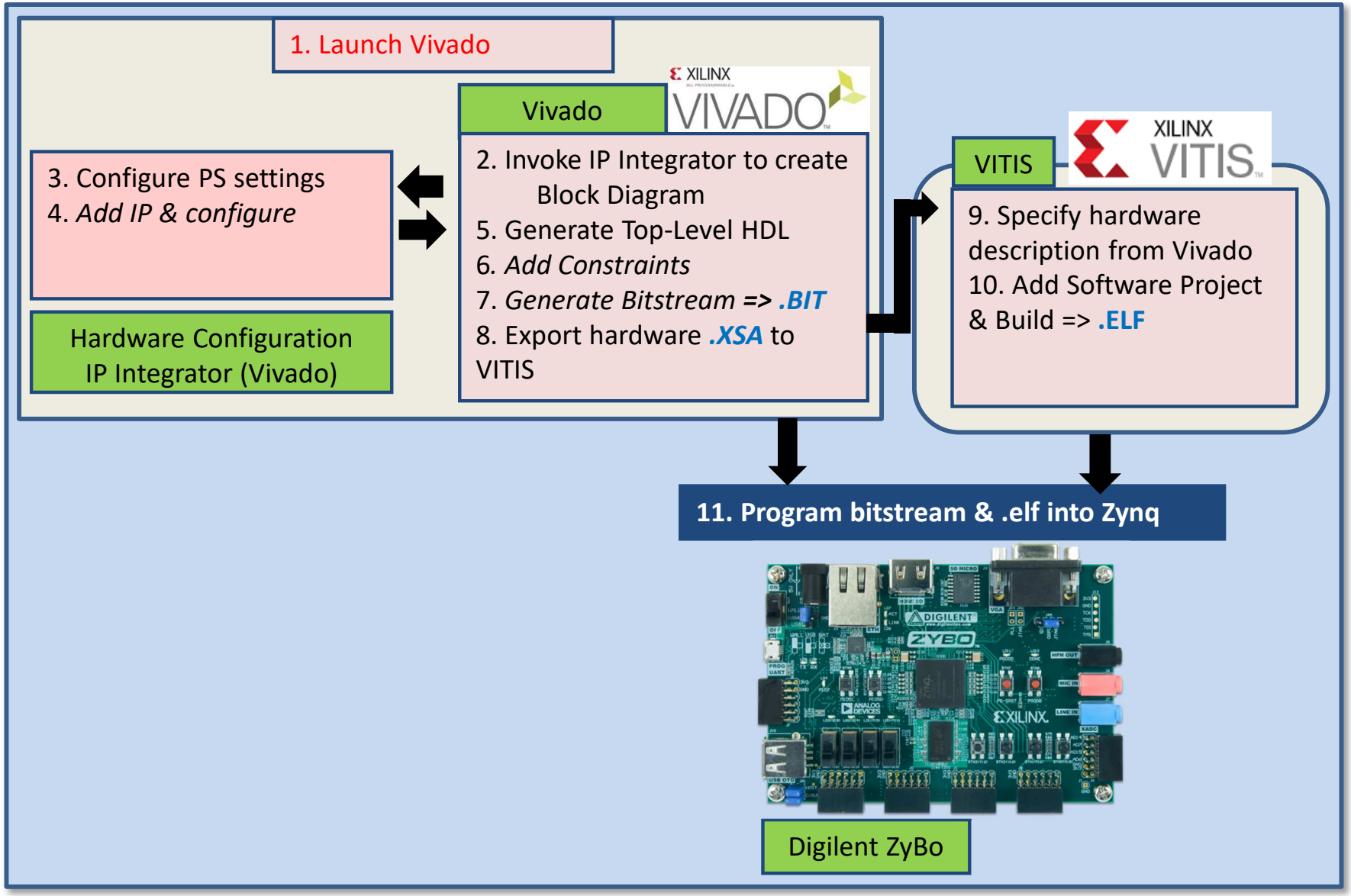
XILINX VIVADO



XILINX VITIS

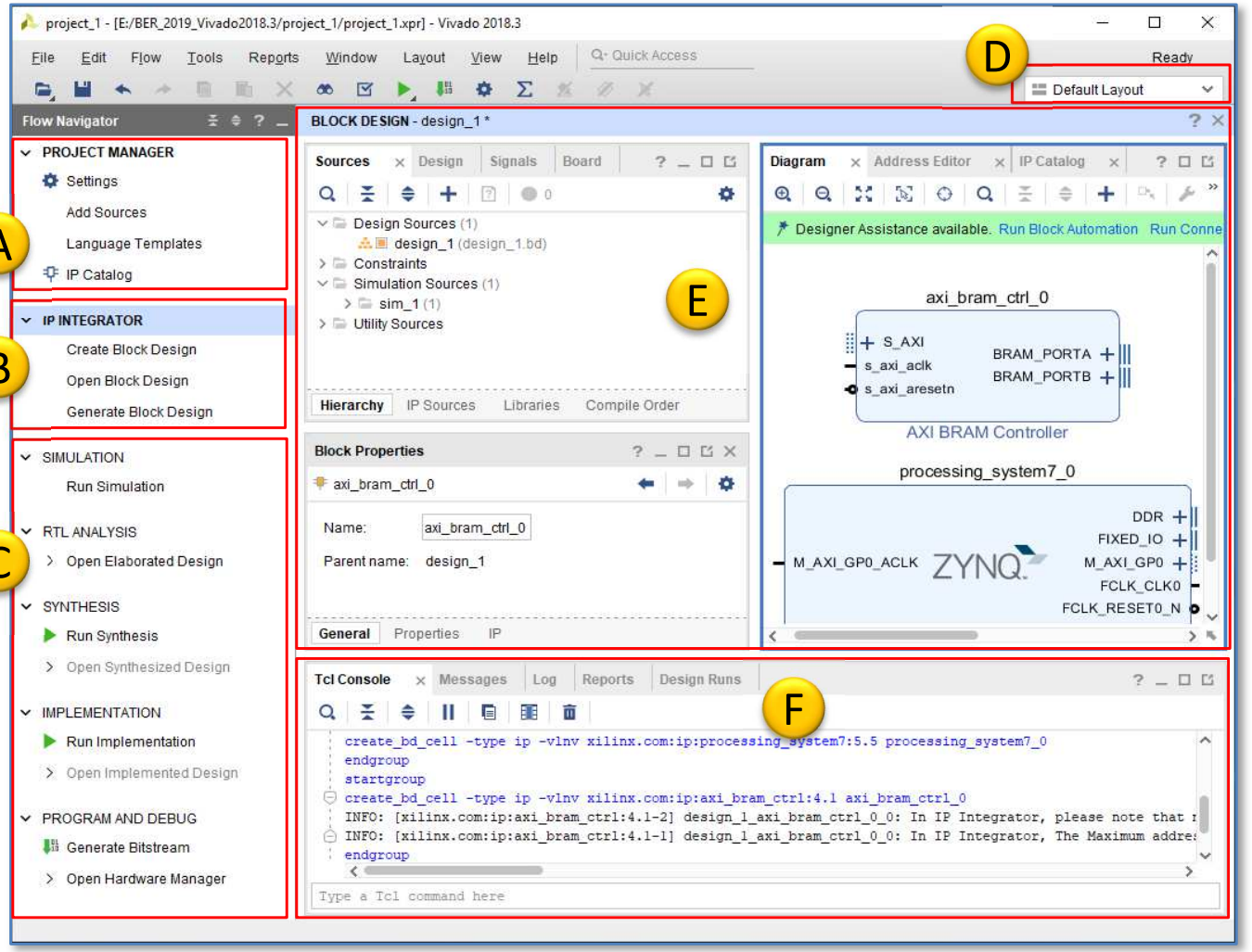
Embedded Systems - Development Flow

A demonstration example:



Vivado – GUI*

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



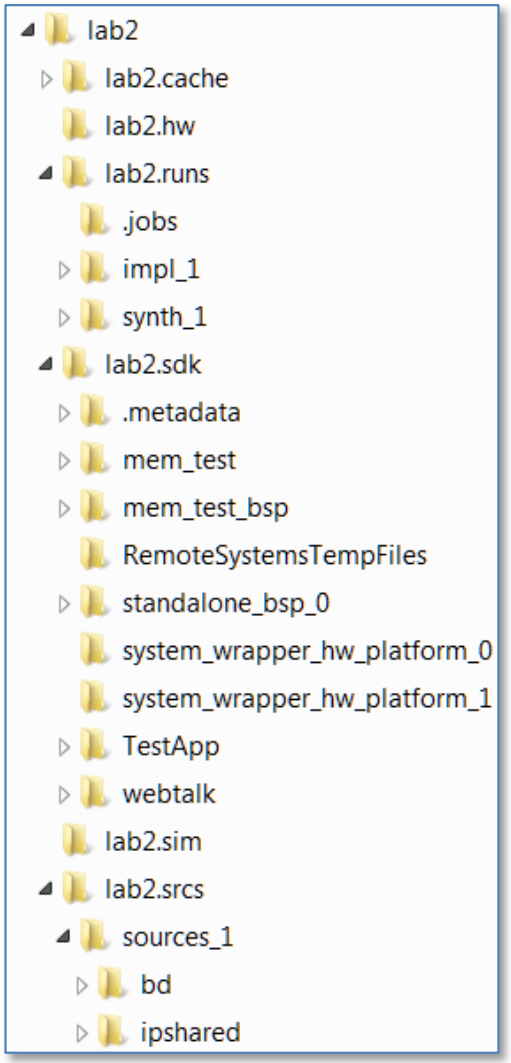
* Note: Vivado can be used in two ways:

- **Project mode**: project management approach with GUI (we use this!)
- **Non-project:** script command approach, without GUI, command line

Vivado - project directory structure

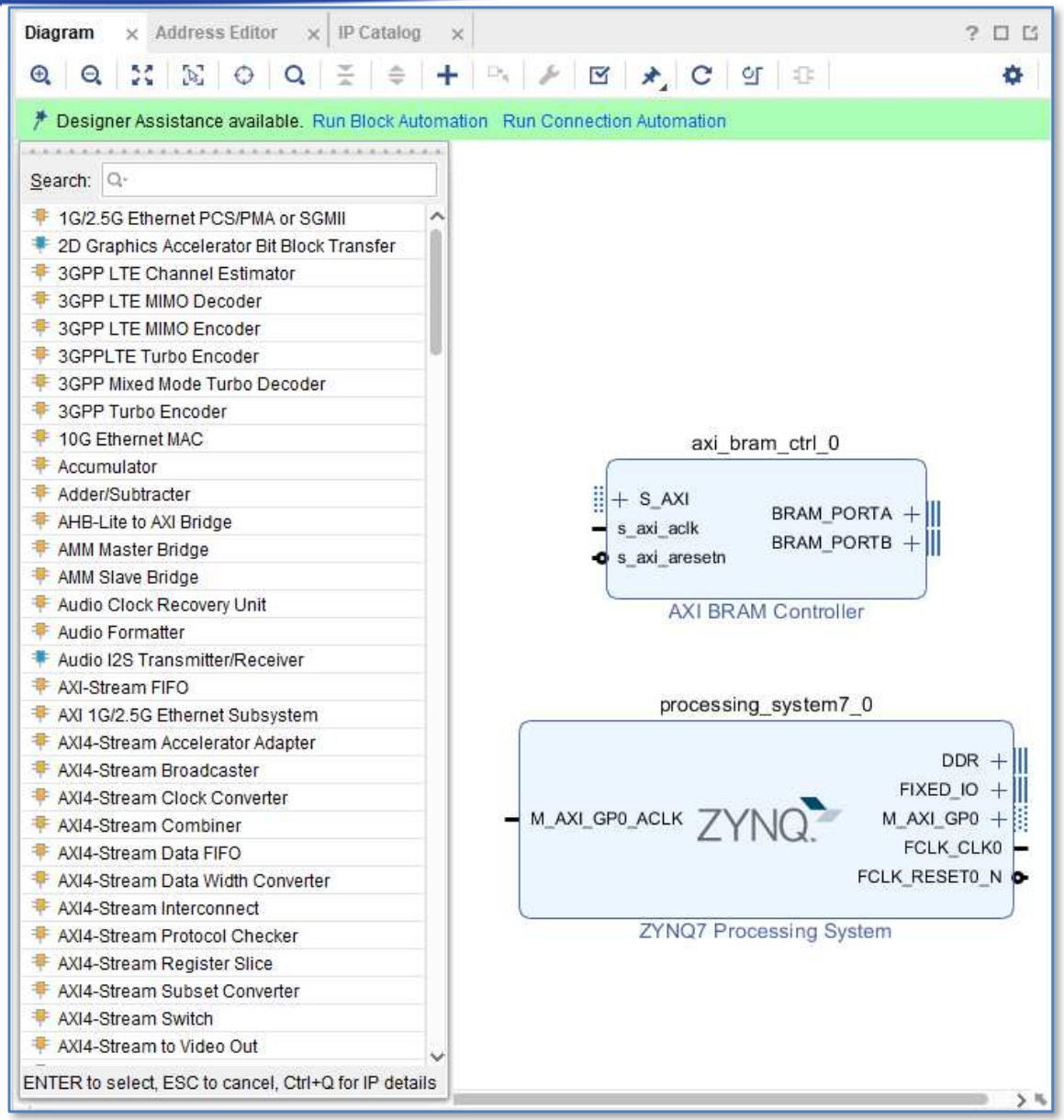
- **in Top-level directory**
  - .xpr = Vivado Project File (xml), log, journal

- /<name>.***srcs***
  - Project source files (BD, XDC), IP Integrator files
- /<name>.***sim***
  - Szimulációs fileok
- /<name>.***runs***
  - szintézis, implementációs futtatások
- /<name>.***sdk***
  - SDK Export directory, Hardware Platform (xml)
- /<name>.***cache***
  - Temporary files

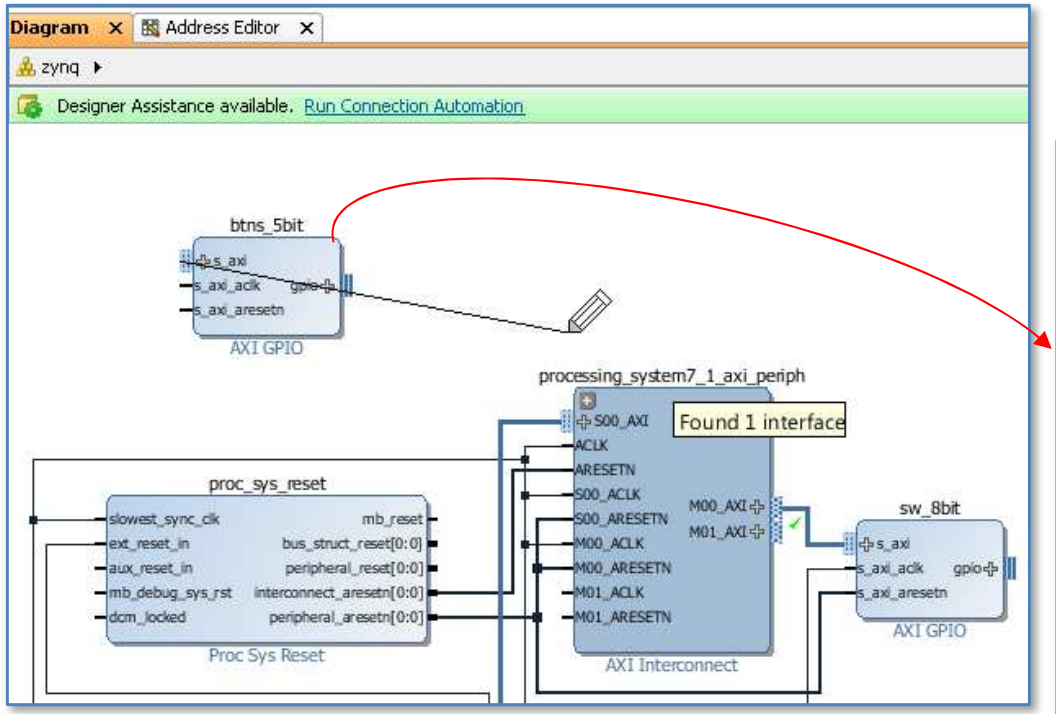


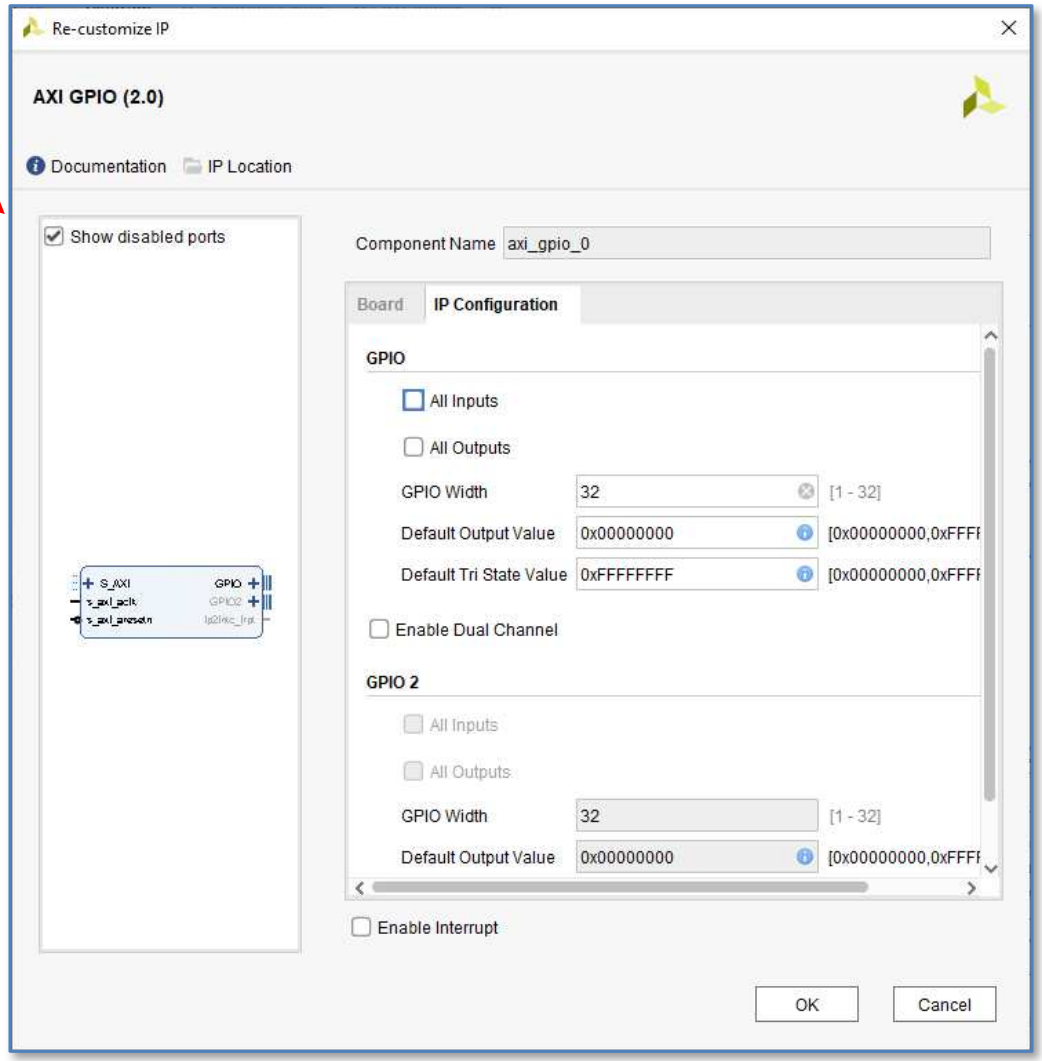
IPI – IP Integrator

- **IP – Intellectual Property**
  – „szellemi termék”
- IP Catalogue
  - Drag&Drop
- Efficient GUI
  - Auto router
  - Redraw
  - Yielding wrong connections
  - Block groupping, hierarchy handling
- IP Packager
  - Import custom, or external partner / 3rd party IPs



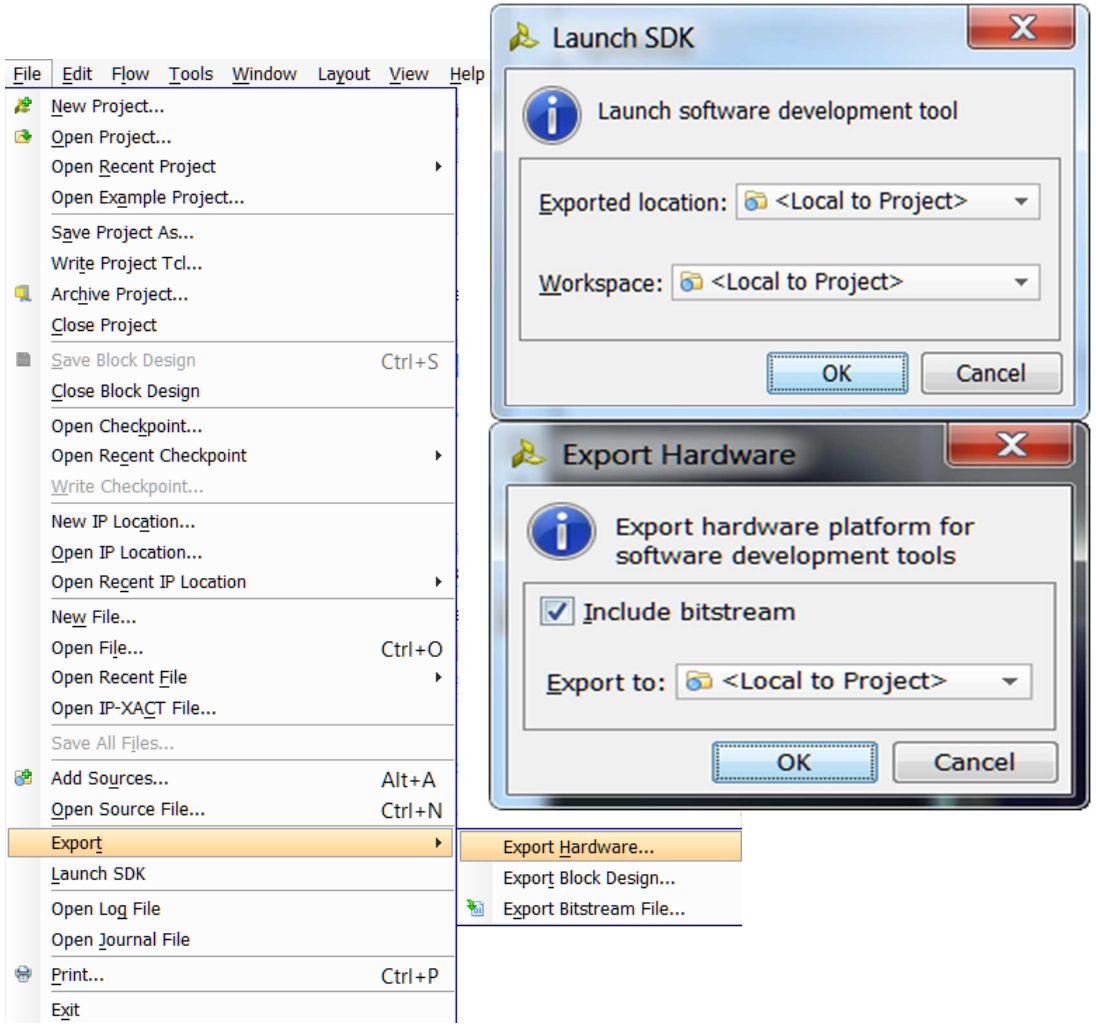
IPI – IP Integrator





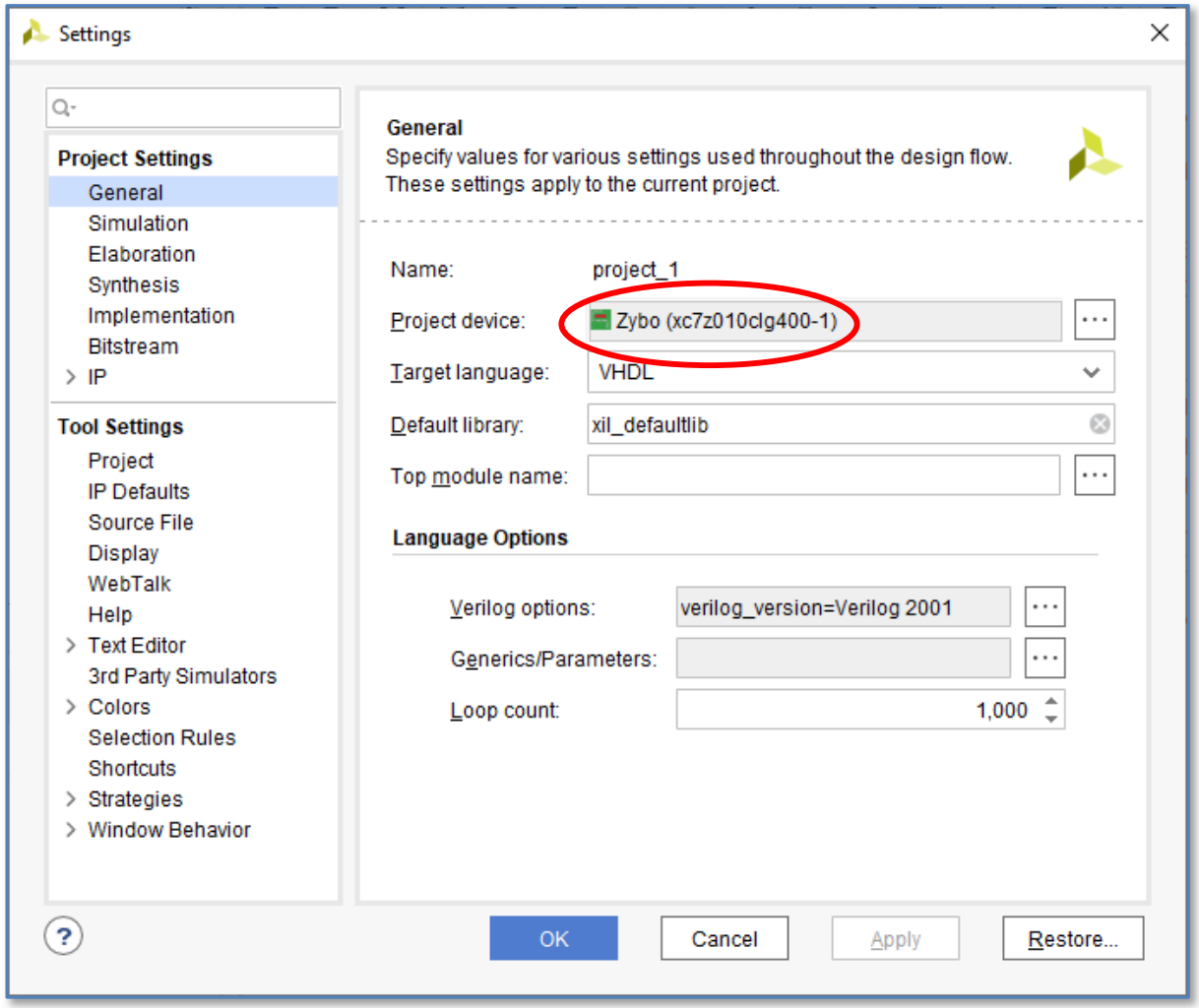
Exporting HW/FW → VITIS

- **.XSA**: Xilinx Support Archive file – HW description of embedded system
- **.BIT:** Bitstream exported (optional, if PL side configured)
- Launch **VITIS SDK** launch



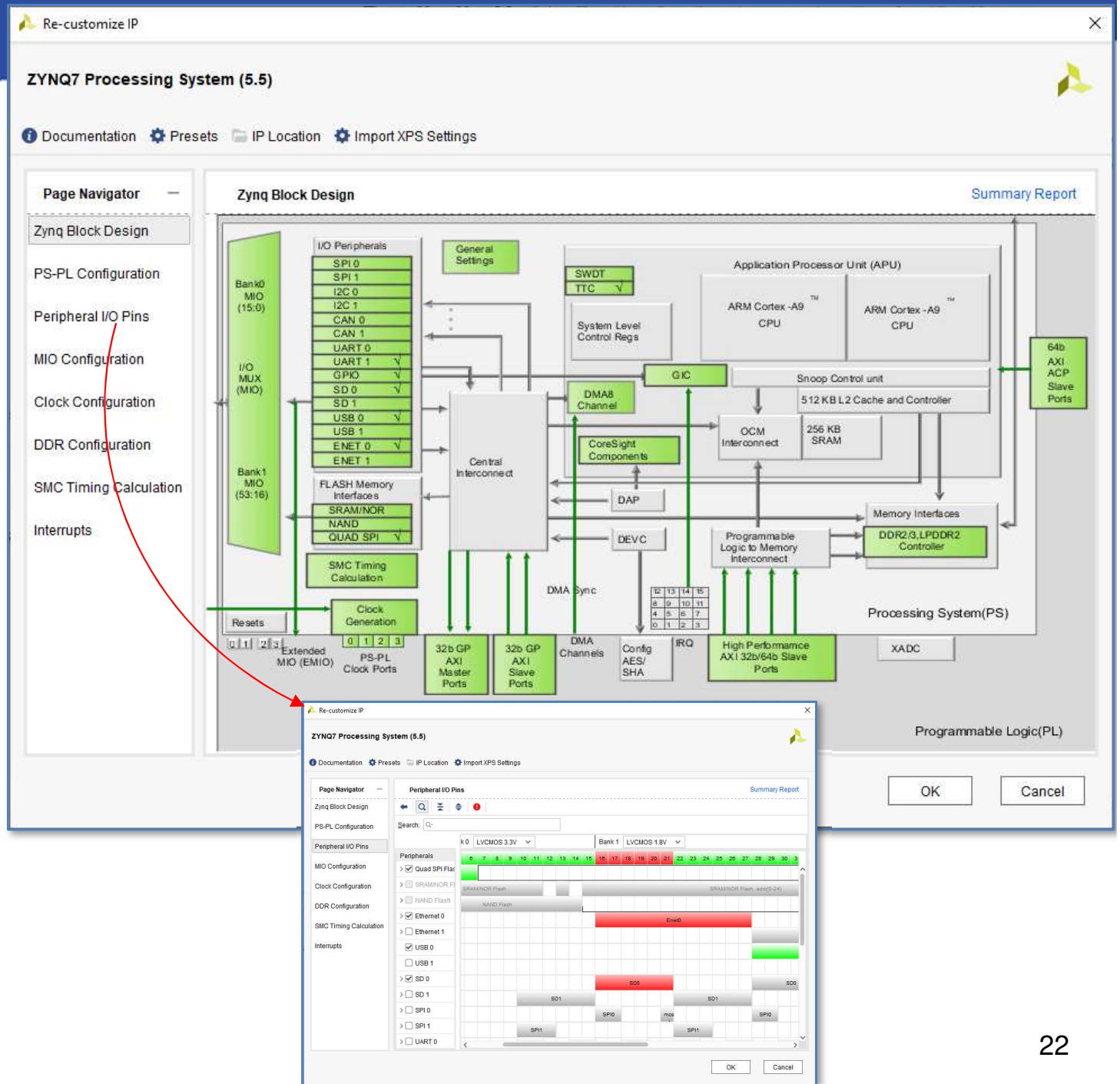
Vivado – Project settings

- Target device settings:
  - FPGA architecture,
  - Device size,
  - Package, pin count,
  - Speed grade
- Simulation, Synthesis, Implementation, Bitstream settings
- IP repository – search path
  - Path for external Ips
- Tool settings



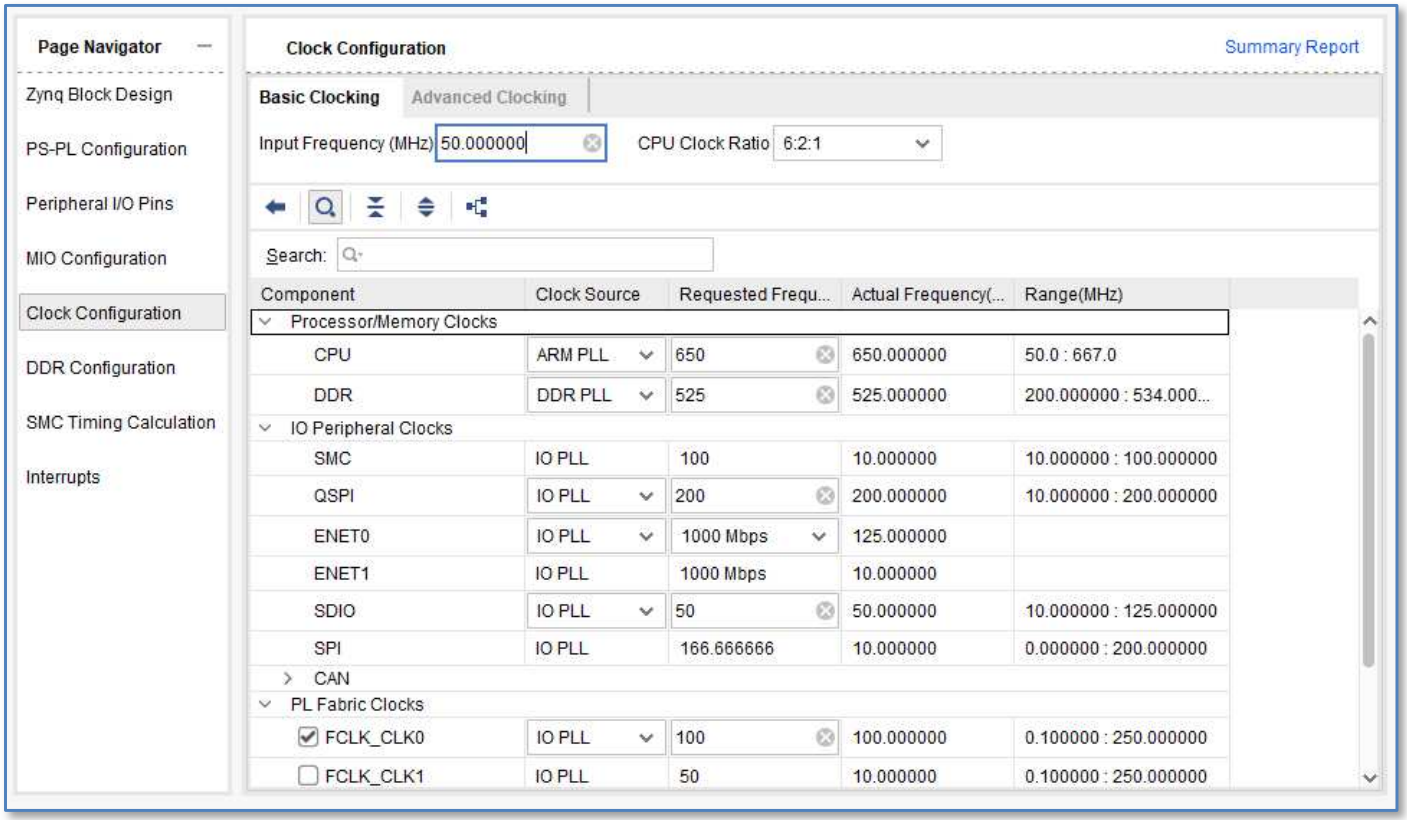
GUI – Zynq PS

- Zynq PS customization
  - ARM cores
  - I/O peripherals
  - Memory system, DDR controller
- I/O partitioning:
  - PS, PL parts
  - MIO configurations
- MemoryMapped configuration registers



Clock configuration

- Clock Configuration (source from ARM_PLL, DDR_PLL, IO_PLL)
  - Changeable Input frequency (Processors, DDR)
  - Frequency for all IO Peripherals can be set
  - Enable/disable PL-side clocks
  - Set Timers



Abbreviations

- Major files (extensions) used or generated by Vivado IPI (FW) embedded system development tool:
  - **.XPR** = Xilinx Project File
  - **.XSA** = Xilinx Support Archive file (XML descriptor → input for VITIS SDK)
  - **.XDC** = Xilinx Constraint File (~ Synopsalapúis .SDC)
  - **.BIT** = Bitstream (FPGA configuration) file

- VITIS SDK (SW) fejlesztő eszköz által generált fontosabb fájlok:
  - **.ELF** = Extensible and Linkable Format
  - **.LD** = Linker Script Download file
  - **.BSP** = Board Support Package (Domain = SW app + OS)

MicroBlaze

MICROBLAZE™ EMBEDDABLE / SOFT PROCESSOR CORE

Brief description – NOT DISCUSSED

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Strukturális és Beruházási Alapok

BEFEKTETÉS A JÖVŐBE

References

📖 MicroBlaze soft-processor

- https://www.xilinx.com/products/design-tools/microblaze.html

📖 MicroBlaze Processor reference guide (Vivado 2020.1):

- https://docs.xilinx.com/v/u/2020.2-English/ug984-vivado-microblaze-ref

📖 MicroBlaze quick start with VITIS:

- https://www.xilinx.com/support/documentation/quick_start/microblaze-quick-start-guide-with-vitis.pdf

📖 MicroBlaze – Design Hub:

- https://docs.xilinx.com/v/u/en-US/dh0020-microblaze-hub

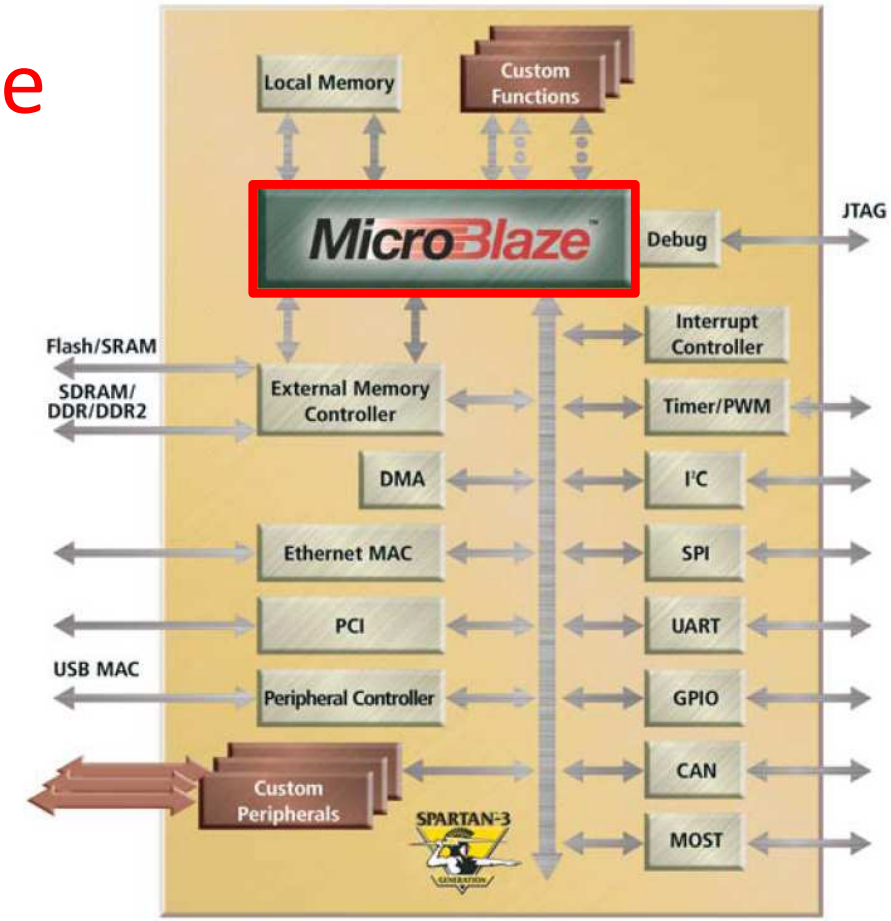
Processor cores on Xilinx FPGA/APSoC

- „Embed***dable"*** = „soft-core" processor = variable/non-dedicated
  - Xilinx *PicoBlaze*: 8-bit (build up from VHDL, Verilog HDL sources)
  - **Xilinx *MicroBlaze*: 32-bit (VITIS/Vivado SDK support!**)
    - It can be connected to AMBA-AXI in Vivado or PLB/OPB (deprecated) in former EDK dev. systems
  - 3rd Party: non-Xilinx manufactured (HDL source)
    - Pl. www.opencores.org
    - NEW ARM Cortex M1/M3 (2019): licensable cores (HDL source)
- Embed***ded*** = „hard-core" processor = dedicated
  - IBM *PowerPC* 405/450 processor 32-bit (deprecated)
    - Exclusively in Virtex-II Pro, Virtex-4 FX, Virtex-5 FXT FPGAs!
  - *ARM Cortex-A9* processor (32-bit): connected to the ARM AMBA-AXI interface
    - Xilinx Zynq APSoC-n integrated ARM cores
  - ARM Cortex A53/A72 (64-bit)
    - Xilinx Zynq APSoC, UltraScale+ MPSoC
  - ARM Cortex-R5 – real-time (32-bit)
    - Xilinx Zynq Ultrascale RFSoC – radio chips

MicroBlaze – Soft-core processor

"Embeddable" processor core

- **RISC instruction set architecture**
- **32-bit soft processor core**
- **133+ MHz clock signal (PLB* / AXI bus)**
- **Harward memory architecture**
  - Instruction Cache / Data Cache
- Low power consumption: ~ mW / MHz
- 3-/5-stages of data line pipe-line
- #32 32-bit general purpose registers
- Timing options (timer/counter)

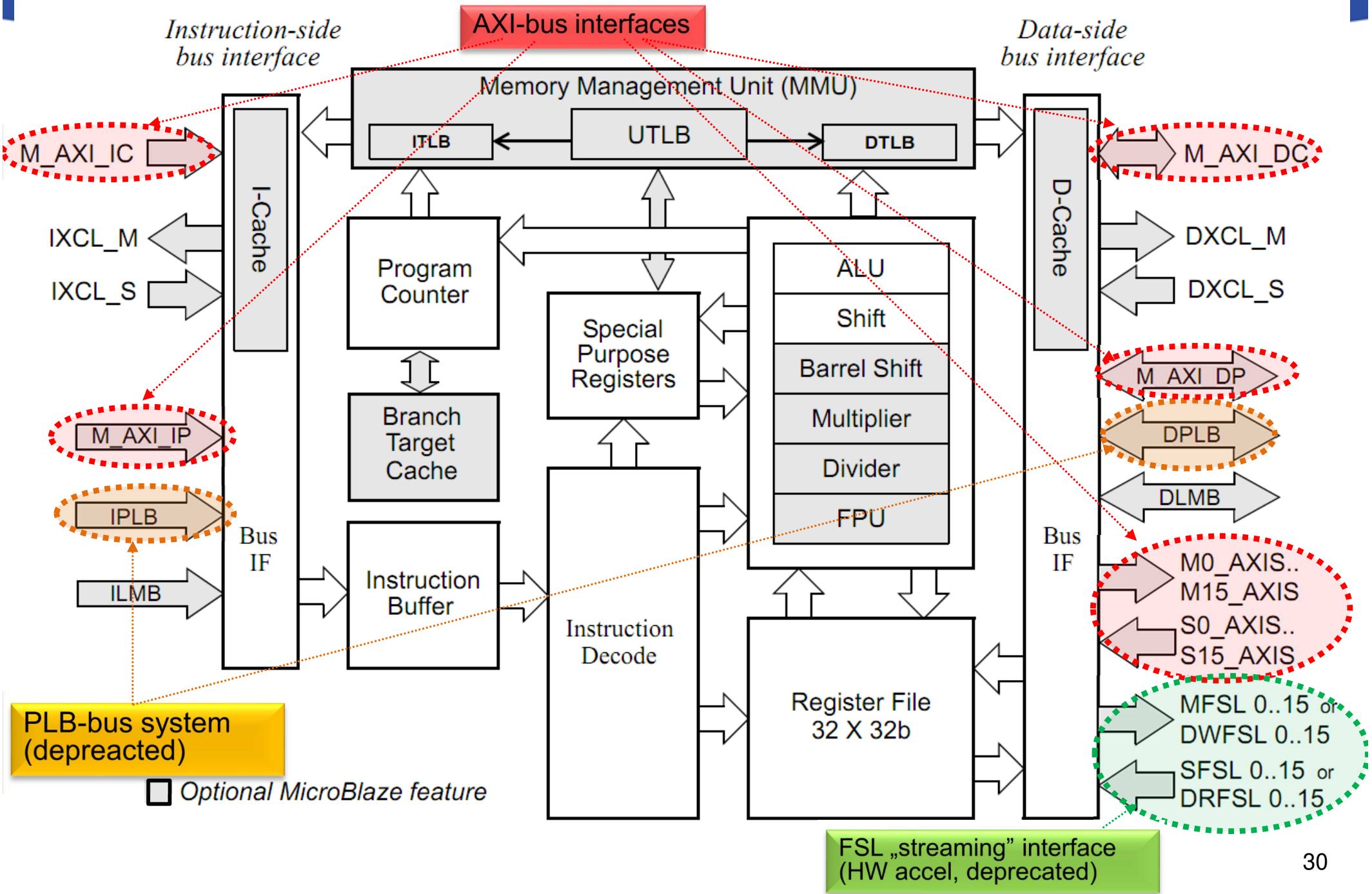


- Many peripherals, communication interfaces can be connected (IP cores)
- It can be implemented on **any** Xilinx FPGA that has sufficient programmable logic resources and is supported by the development software!

MicroBlaze - Performance

- Instruction set (#85): run over 1 clock cycle with except of:
  - Load & Store (2 cycles): memory access
  - Multiplication (2 cycles)
  - Branches (1-3 cycles)
  - 🕮 **Details: \doc\microblaze.pdf** (installed DocNav directory)
- Operatonal frequency – fast speed grade, 5 stage-pipeline
  - 307 MHz on the Virtex-6 (-3) FPGA
  - 154 MHz on the Spartan®-6 (-3) FPGA
  - 119 MHz on the Spartan-3 (-5) FPGA
- Computing performance 1.15 **DMIPS**/MHz (=**Dhrystone MIPS** means a raw program-instruction execution speed/ sec)
- Resource utilization – num of LUTS in case of area vs. speed optimized version
  - 779/1,134 LUTs in the Virtex-6 FPGA (LUT-6)
  - 770/1,154 LUTs in the Spartan-6 FPGA (LUT-6!!)
  - 1,258/1,821 LUTs in the Spartan-3 FPGA (LUT-4!!)

MicroBlaze - Microarchitecture



MicroBlaze – general features

„Embeddable” processor core

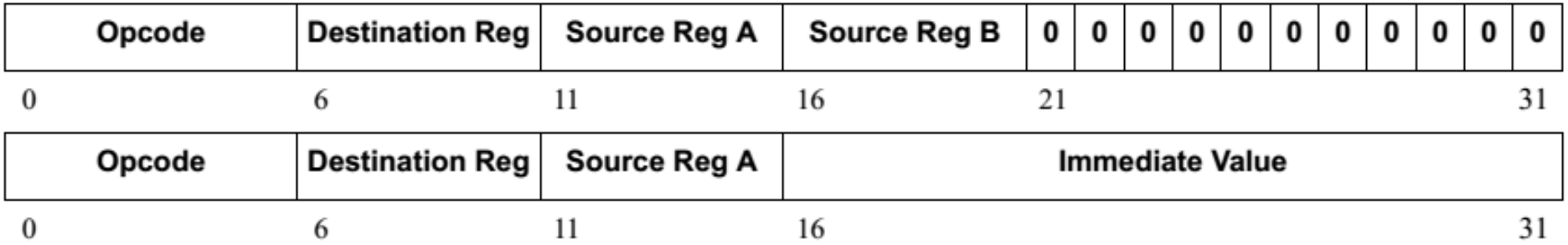
- **32-bit soft processor core**
- **RISC instruction set** architecture
  - ~ #85 instructions and their subtypes
- **100+ MHz** scalable clock (depending on FPGA architecture and bus system: AXI vs. older PLB)
- **Harvard** architecture
  - Instruction Cache / Data Cache
- Low power consumption: ~ mW / MHz
- #32 32-bit General Purpose Registers
- Timing options (timer/counter)
- Many types of peripherals, communication interface can be connected (in form of IP cores)
- It can be implemented on any Xilinx FPGA that has sufficient resources and is supported by the development software!

MicroBlaze – Parts I.

- *Flexible 32-bit processor core*
  - Current latest version v.11.0 (on Xilinx Vivado 2020.1)
  - Configurable 3-stage (resource optimized) or 5-stage data line pipeline (performance optimized version)
- Optionally configurable **instruction** and **data** cache (**I**-Cache, **D**-Cache)
- Direct mapping (1-way, group associative cache)
- It has #32 General File registers (32-bit each) (Register File)
- **Scalable 32-, 64-, or 128-bit data bus**
- **32-bit address / instruction bus**

MicroBlaze – Parts II.

- **32-bit instruction bus** (supports up to *3-address* instructions, and *two different addressing modes*: *Register* and *Register-Imm*)



  - arithmetic,
  - logic,
  - Load & Store (Memory register for transfer operations - RISC feature),
  - Program organizer instructions (e.g. Branch - conditional execution),
  - Special instructions.

MicroBlaze – Parts III.

- optional **Memory Management and Memory Protection Unit (MMU)**
  - Required to run an embedded Linux OS on the MicroBlaze core with advanced memory management
- optional processor core **floating point arithmetic unit (FPU)**
  - Support for standard IEEE-754 format

MicroBlaze – Parts IV.

- **Barrel Shifter**: arbitrarily scrollable register (optional),
- **Hardware multiplier** (optional): when used, uses 3 MULT18x18 / DSP dedicated blocks.
  - This is a platform specific operation unit: multiplying the 32 × 32-bit operands of the processor → with a 64-bit result,
- **Hardware divider** circuit (optional): includes multiplier circuits, logic cells and possibly dedicated BRAM cells,
- **Hardware-supported debugging** (with extensive debug function)
  - Xilinx MDM - MicroBlaze Debug Module,
  - Xilinx ILA, embedded logic analyzer module (s) for professional external measuring instruments
  - supported by another manufacturer (e.g. Agilent Trace Module)

MicroBlaze – Parts V.

- It basically follows the Big-Endian, i.e. bit / byte-reversed storage format (but since the EDK 13.x system, endianity can be configured in the two different bit / byte orders):
  - **Big-endian**: to support I/O peripherals that can be connected to the previous PLB (Processor Local Bus) bus
  - **Little-endian** (optional): to support the latest ARM hard-core / MicroBlaze soft-core AXI bus systems (in the latter case configurable)

ARM™ EMBEDDED / HARD PROCESSOR CORE

General description

MAGYARORSZÁG KORMÁNYA

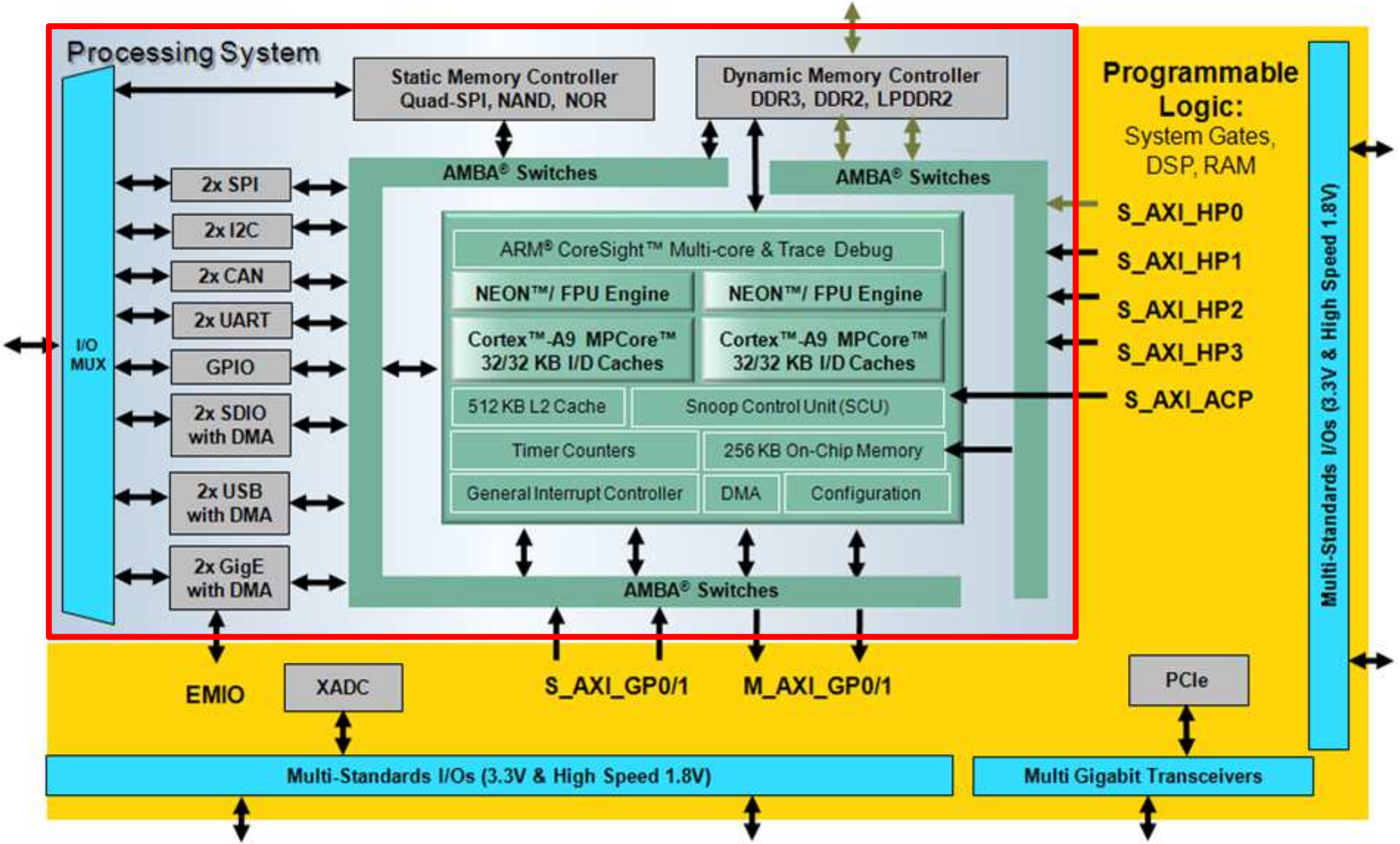
Európai Unió
Európai Strukturális és Beruházási Alapok

BEFEKTETÉS A JÖVŐBE

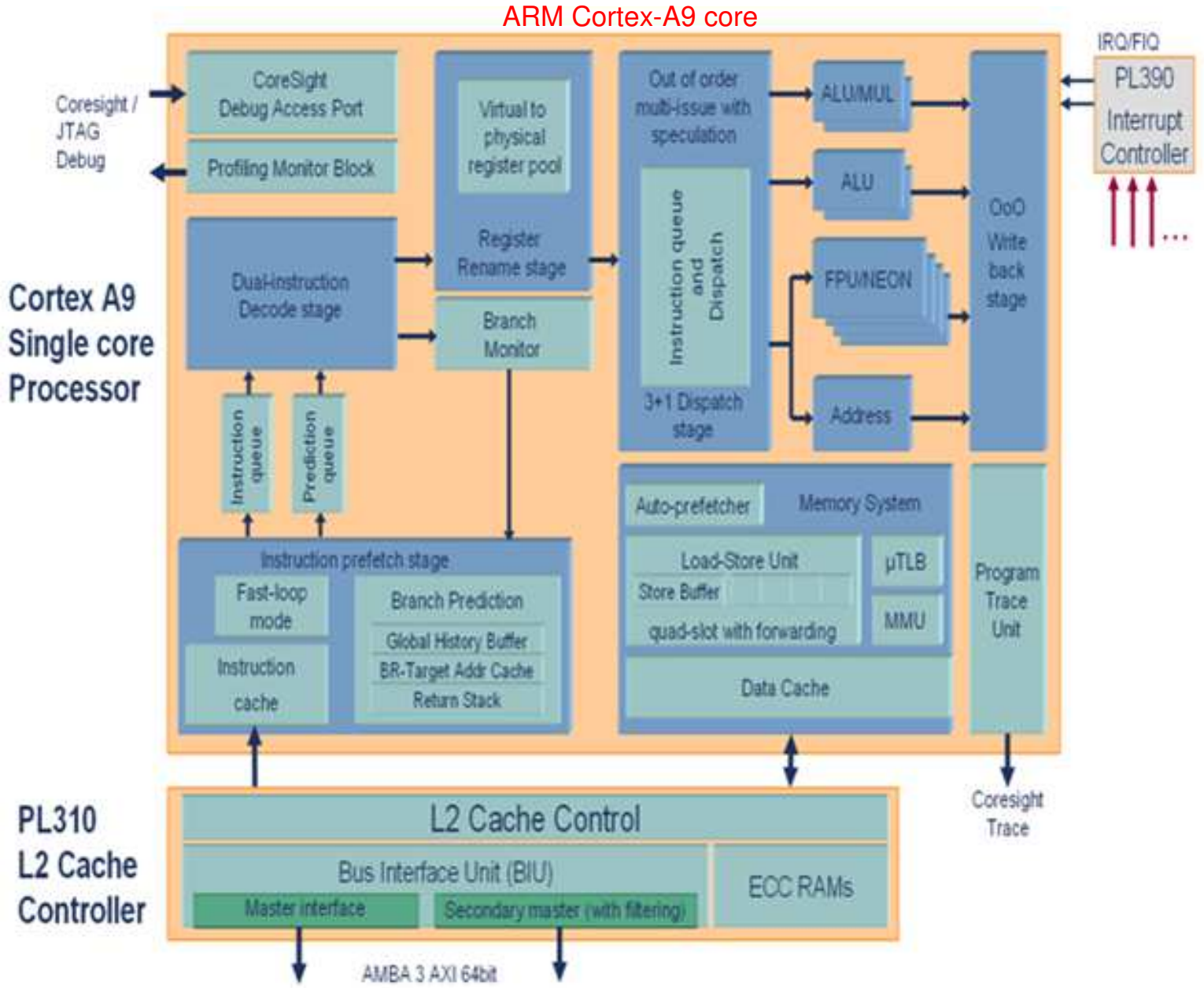
ARM processor architecture I.

- Integrated into Zynq PS side
- ARM Cortex-A9 processor based on **ARMv7-A architecture**
  - ARMv7 - ARM Instruction Set Architecture (ISA)
  - ARMv7-A: Application Memory Management Unit (MMU)
    - Called also as APU (Application Processor Unit)
  - ARMv7-R: Real-time Memory Protection Unit (MPU)
  - ARMv7-M: Microcontroller: M1, M3 (since 2019*, free but license required)
- ARMv7 ISA - instraction set architecture
  - 16 bits/32 bits
  - NEON: ARM's Single-Instruction Multiple-Data (SIMD) extension
- ARM Advanced Microcontroller Bus Architecture (AMBA®) protocol
  - AXI3: v3 ARM „bus" interface
  - AXI4: v4 (extended bursts)
- „Cortex" newest processor families: A9, A53, A72, and R5 (radio)

Zynq APSoC architecture

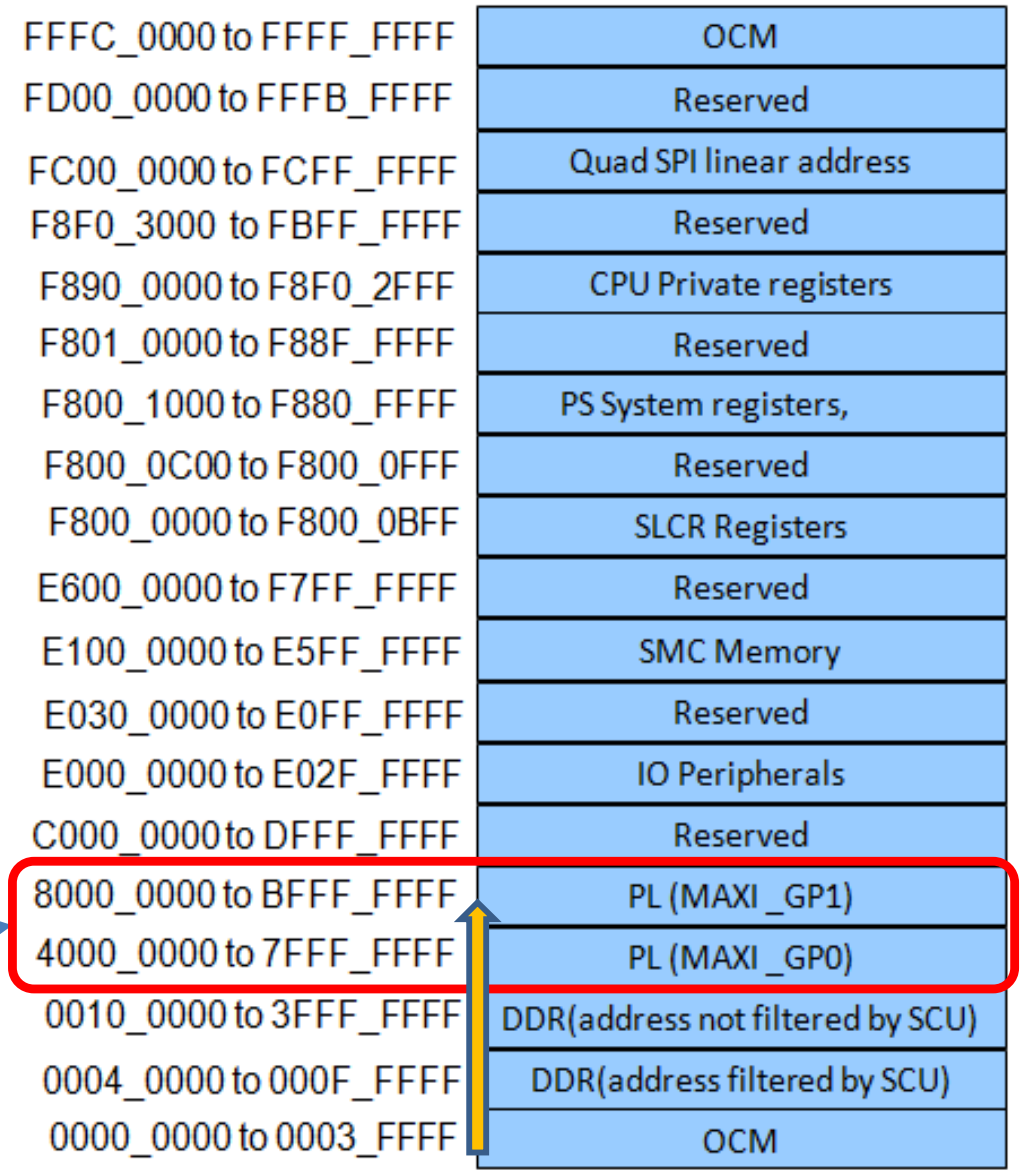


ARM processor architecture II.



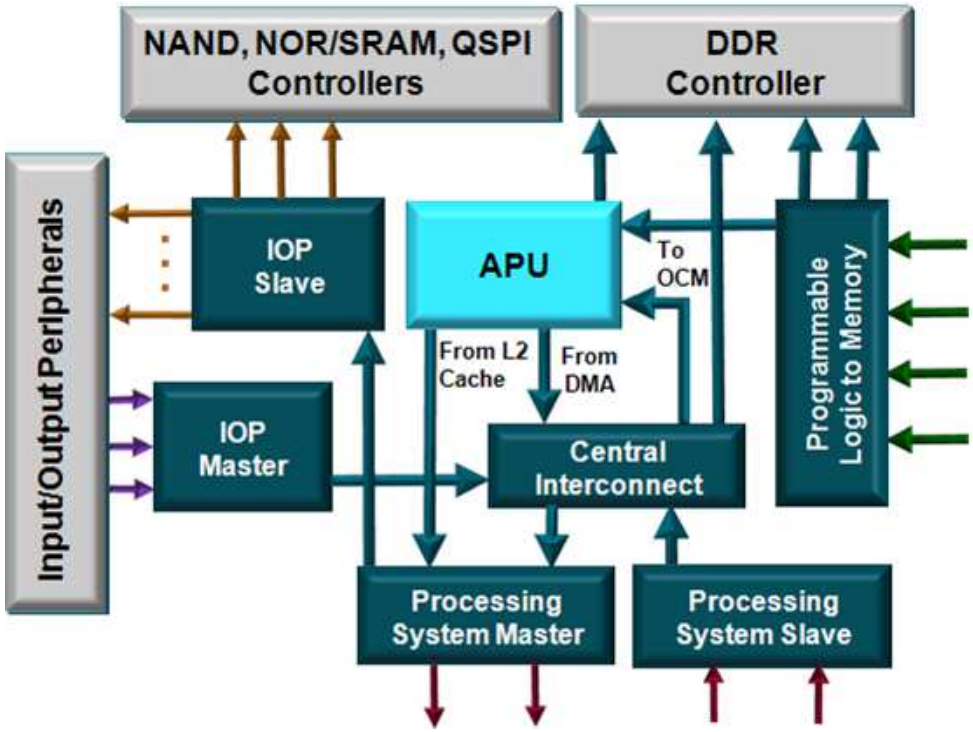
ARM PS (PL) memory map

- 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**)



Important parts of PS

- Application processing unit (APU)
- I/O peripherals (IOP)
  - Multiplexed I/O (MIO), extended multiplexed I/O (EMIO). Max 54 pin!
- Memory interfaces
- PS interconnect
- DMA
- Timers
  - Public and private
- **G**eneral **i**nterrupt **c**ontroller (GIC)
- **O**n-**c**hip **m**emory (OCM): RAM
- Debug controller: CoreSight

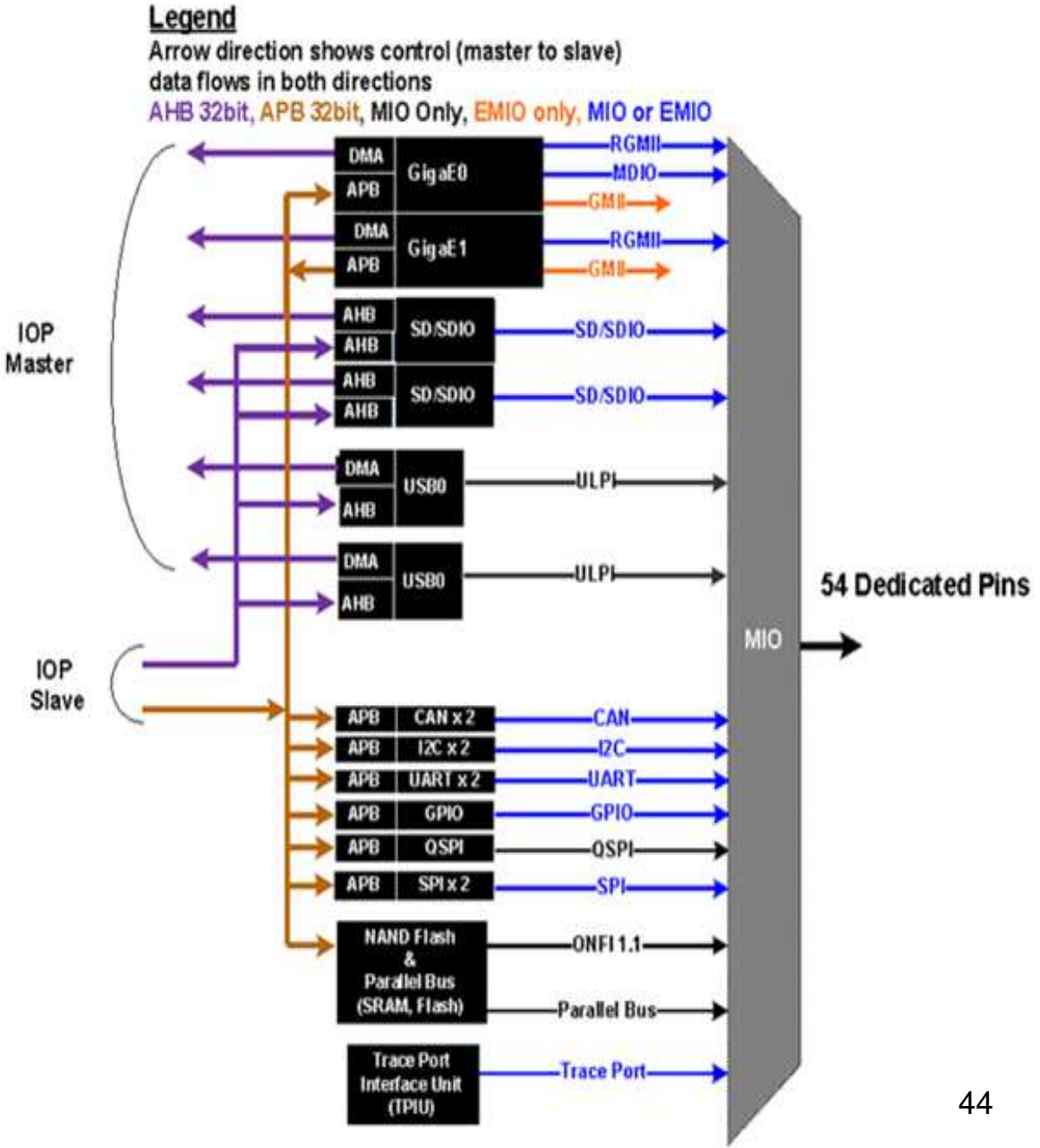


Boot order

- CPU0 boots from OCM ROM; CPU1 goes into a sleep state
- On-chip boot loader in OCM ROM (Stage 0 boot)
- Processor loads **F**irst **S**tage **B**oot **L**oader (**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

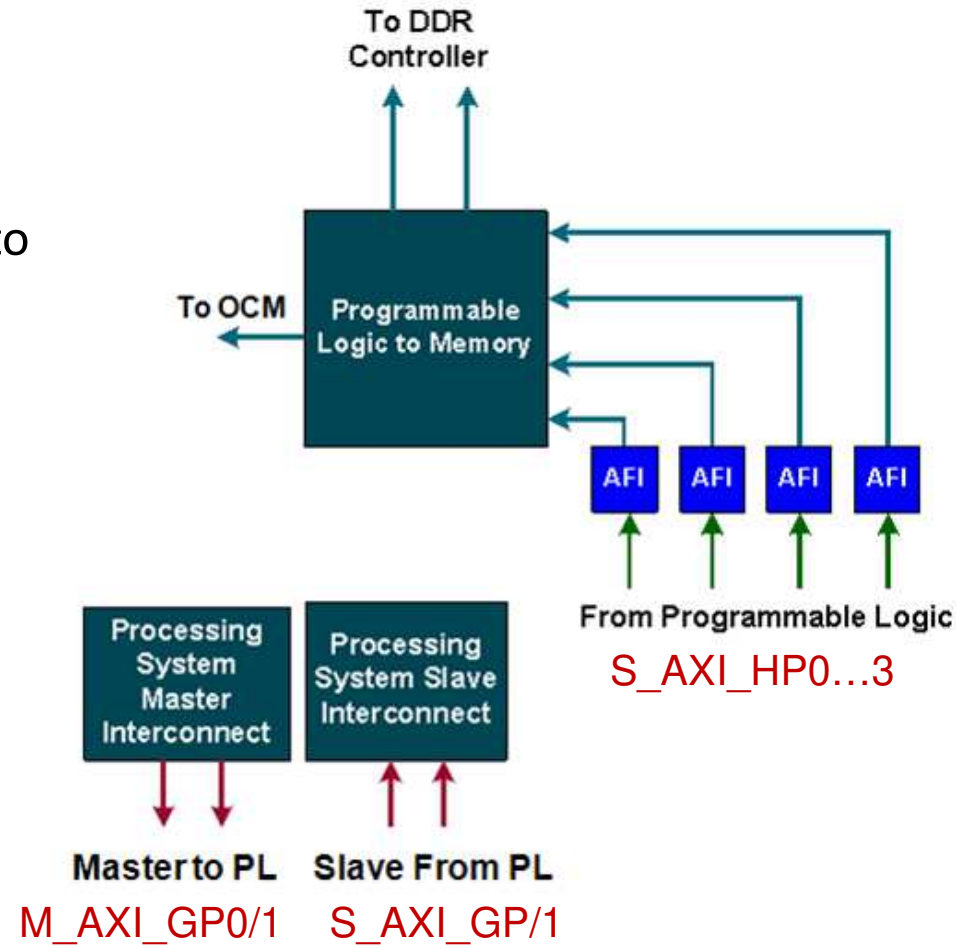
IOP – IO peripherals

- 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 interconnetions („bridges”)

- **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



BUS INTERFACES

General description

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Strukturális és Beruházási Alapok

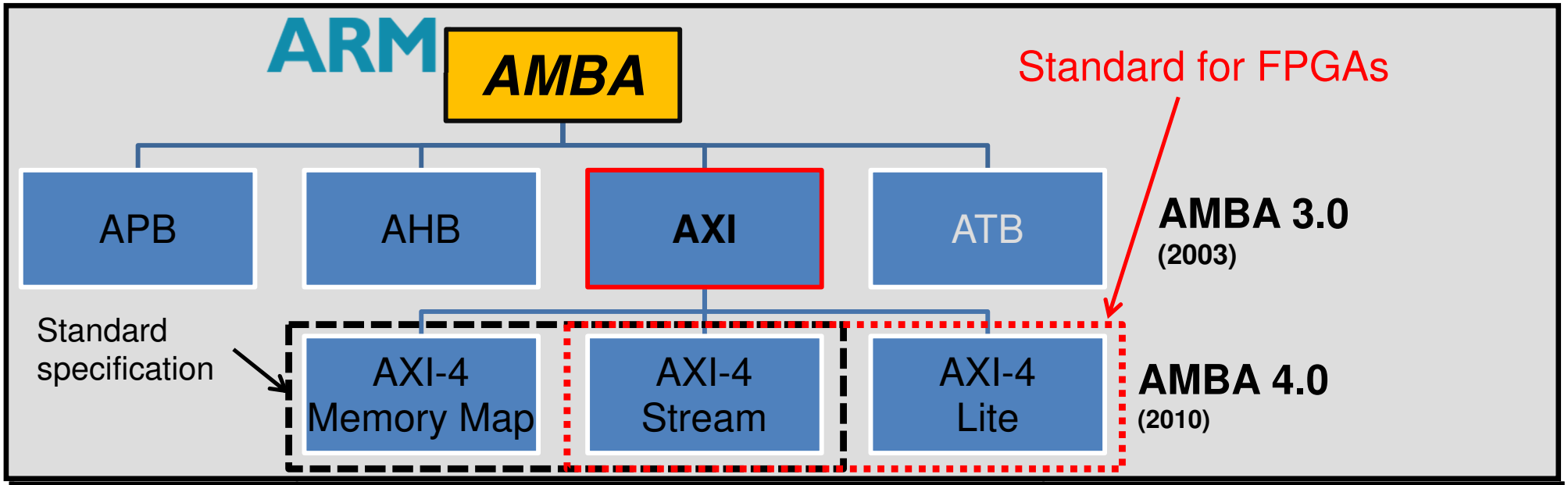
BEFEKTETÉS A JÖVŐBE

Buses – „1x1”

- Bus master (**M**): initiate a bus transaction (R/W)
- Bus slave (**S**): only responds to a bus request
- Bus arbitration (**A**): 3-step decision process
  - The master (M) device sends its bus reservation request to the Arbitrator (A) via the BR (bus request) line for the next transaction
  - The Arbiter (A) continuously monitors (polls) the demands and it sends a BG (bus grant) signal to the master unit (if its priority was the highest from several Masters at the same time)
  - The bus requesting master (M) unit receives the BG signal and then initiates the **transaction** (read-write) to the slave (S) unit when the previous master (M) unit finished its current transaction
- Arbitration = ? decision mechanism. Who should be the Master for the next transaction? The decision can be based on:
  - Fixed-priority, or FIFOed
  - Round-robin, or
  - Hybrid method

ARM AMBA – AXI interface

- AMBA: Advanced Microcontroller Bus Architecture



Interfész	Tulajdonság	Hasonló busz
AXI4-MM: Memory Mapped / Full	**Traditional address/data Burst – „löket”** (SAMD = single address → multiple data). Burst: max 256 unit	**PLBv46, PCI**
AXI4-Streaming	**No address, only data transfer — Burst-like (unlimited burst)**	**Local Link / DSP interfacees / FIFO / FSL**
AXI4-Lite	**Hagyományos cím/adat — Non burst-like** (SASD = single address → single data) Data can be 32 / 64 bits *	**PLBv46-single (deprecated)** **OPB (deprecated)**

AXI – Advanced Extensible Interface

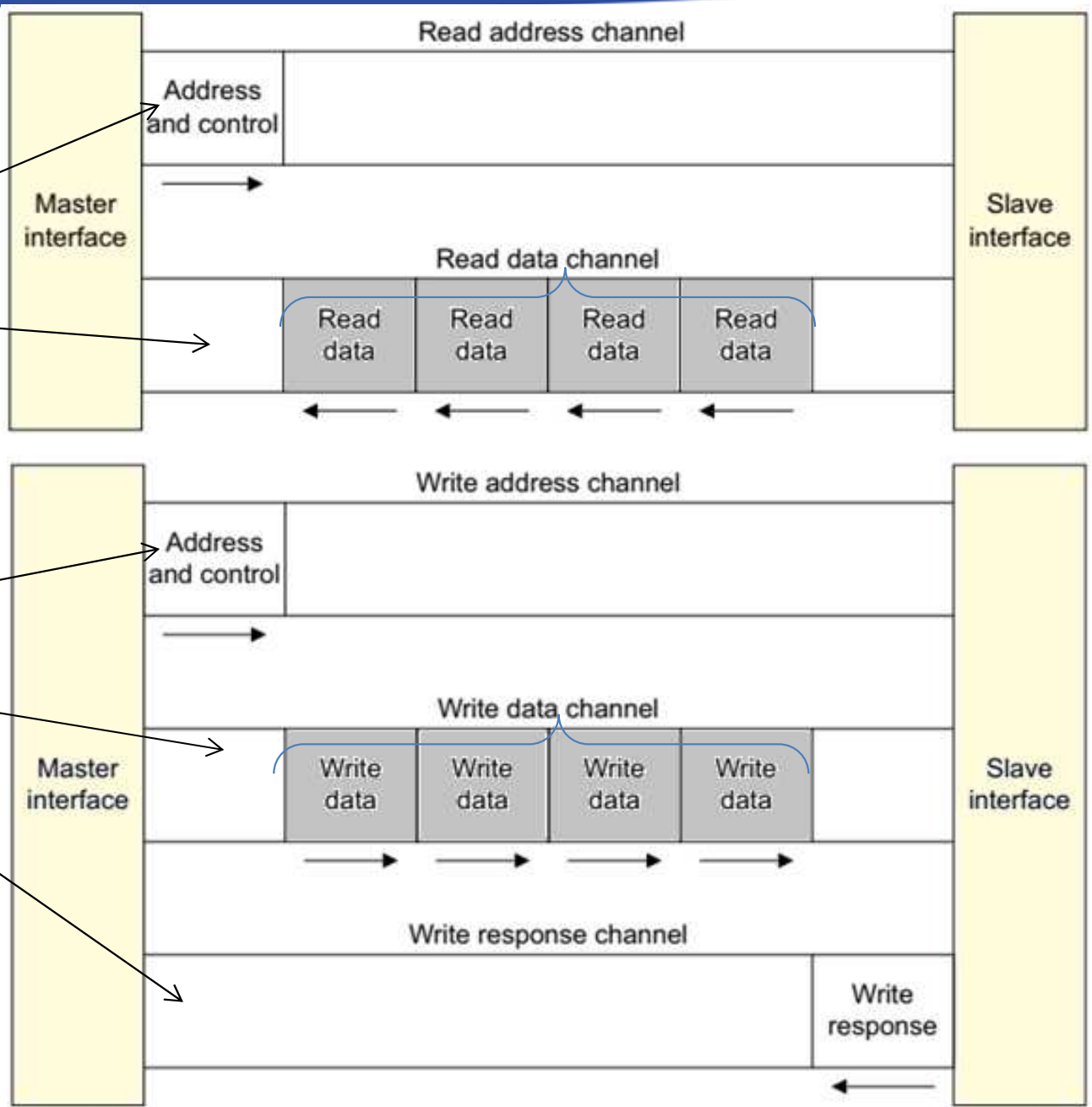
- Standard **interface** and **protocol** definition,
- Widely used,
- Not considered to a „bus” connection!

- The AXI specifications define it more as an „**interface**” (rather than a bus connection!):
  - It does not specify how to connect the IP peripherals in the system, but
  - Defines the inputs and outputs of an IP peripheral.

AXI4 basic mode (memory mapped)

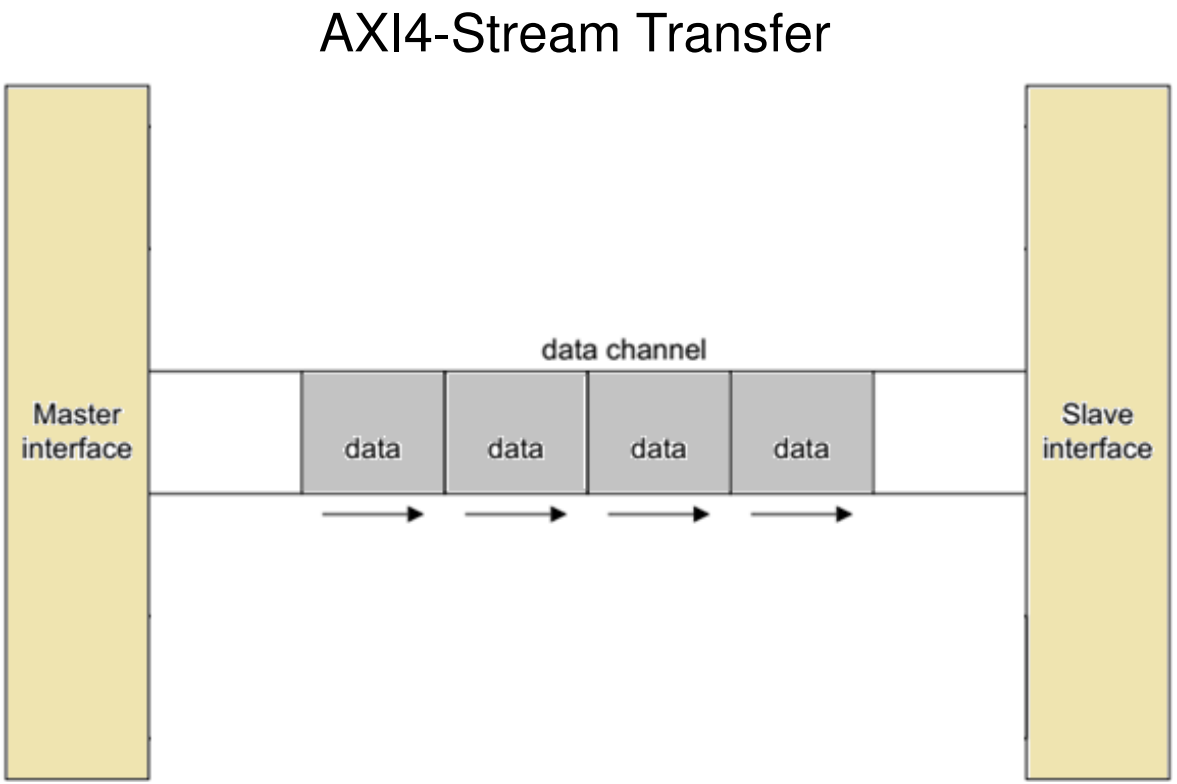
5 communication channels:

1. Read Address Channel
2. Read Data Channel
3. Write Address Channel
4. Write Data Channel
5. Write Response Channel



AXI4 Stream mode

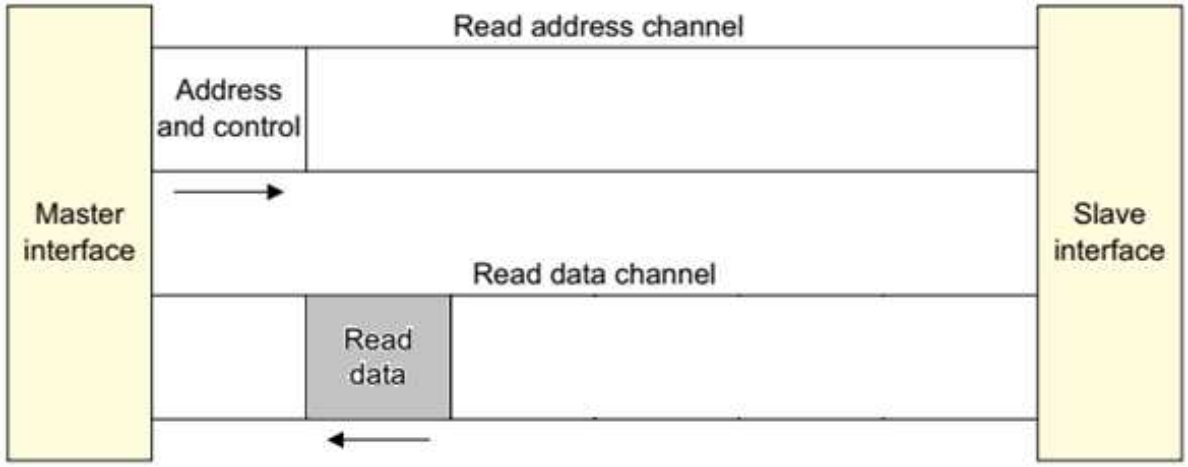
- **No address channel**, no read and write, always just master to slave
  - Effectively an AXI4 “write data” channel
- Unlimited burst length
  - AXI4 max 256
  - AXI4-Lite does not burst
- Virtually same signaling as AXI Data Channels
  - Protocol allows merging, packing, width conversion
  - Supports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



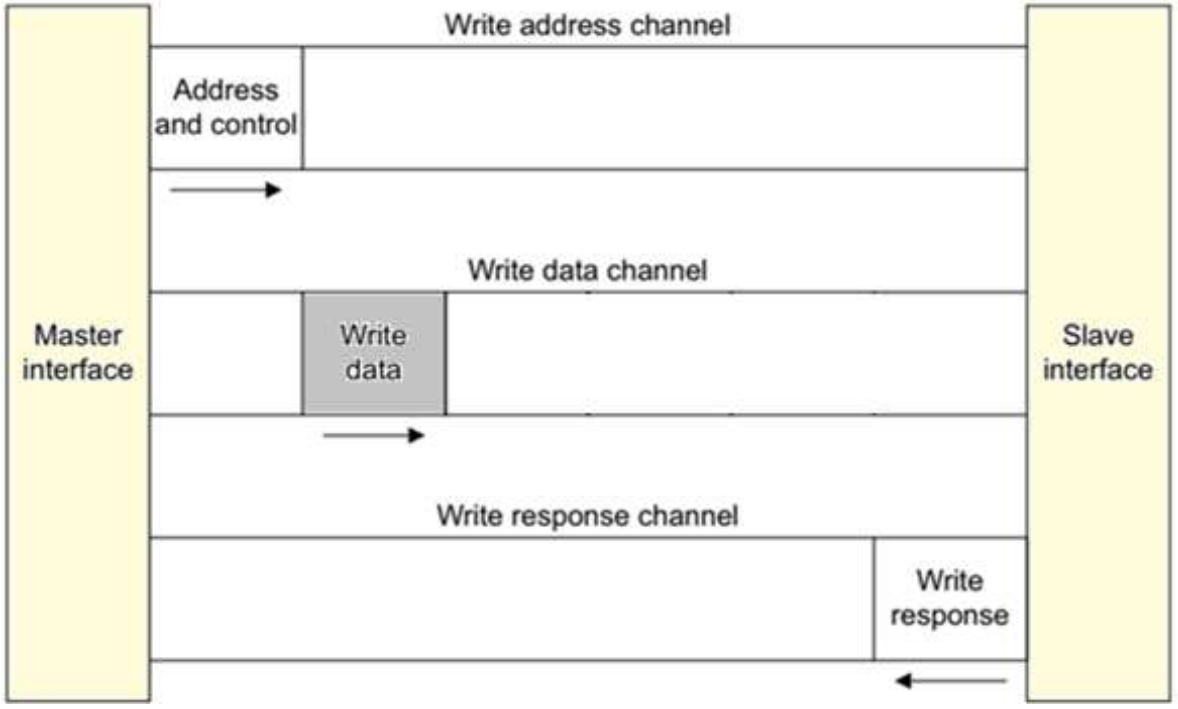
AXI4 Lite mode

- 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)

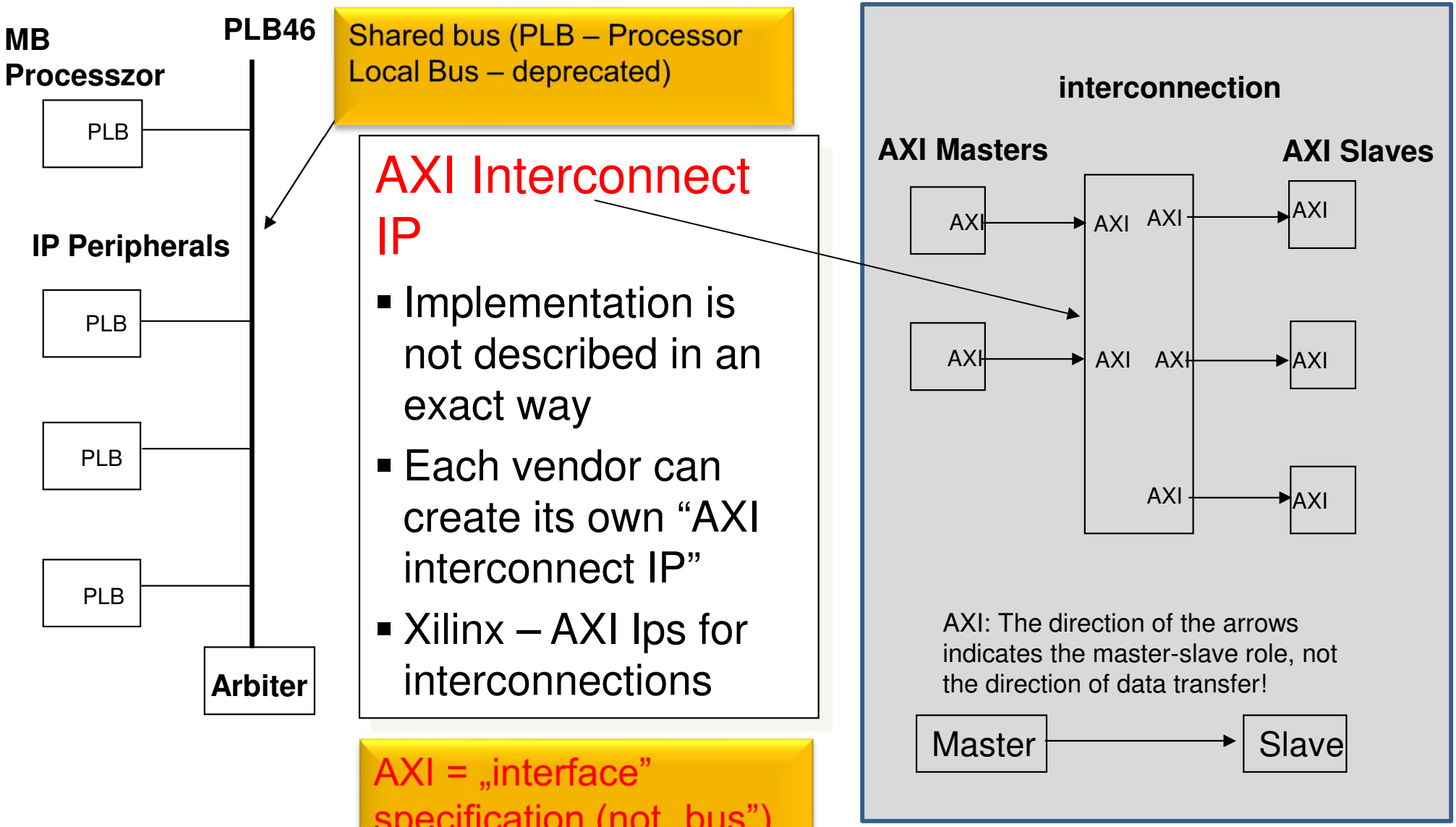
AXI4-Lite **Read**



AXI4-Lite **Write**



AXI as an interface



LMB – Local Memory Bus

- The **LMB** provides data / instruction access to the MicroBlaze processor @ 1 clock cycle
  - on-chip, dual-port memories (BRAMs)
  - own dedicated data / instruction bus
- Simple synchronous protocol
  - **DLMB**: **D**ata interface LMB (BRAM only)
  - **ILMB**: **I**nstruction interface LMB (BRAM only)

Summary – buses & interfaces

- **MicroBlaze/ARM** : AXI interface supported in newest Vivado/VITIS development environment
  - **AXI**: interface for high-speed, point-to-point interconnection
  - MicroBlaze:
    - **AXI:** supports both Little and Big-endian format
- **LMB**: processor's local memory bus (for data, stack and interrupt handler routines) – only for MicroBlaze processors
  - Fix and deterministic latency

EMBEDDED SW DEVELOPMENT ENVIRONMENT (XILINX VITIS)

Brief description

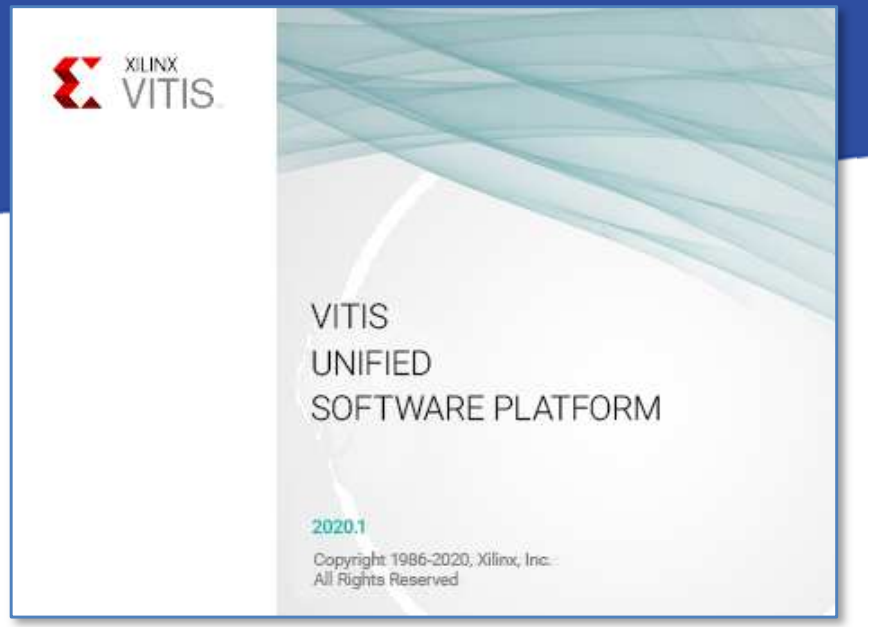
SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Strukturális és Beruházási Alapok

BEFEKTETÉS A JÖVŐBE

VITIS - SDK



Key features:

- Full-featured IDE for SW dev.
  - Eclipse-based, open-source
- Independetly works from Vivado
  - Embedded SW development
- Support Embedded HW/FW system debug
  - Multi-processor platform (MPSoC/APSoC)
  - Multi-BSP / domain SW systems
  - Multi-SW application
- Efficient C/C++ source editor and debugger (navigator)

*Utilization of VITIS SDK will be introduced in slides of BER-04

VITIS SDK

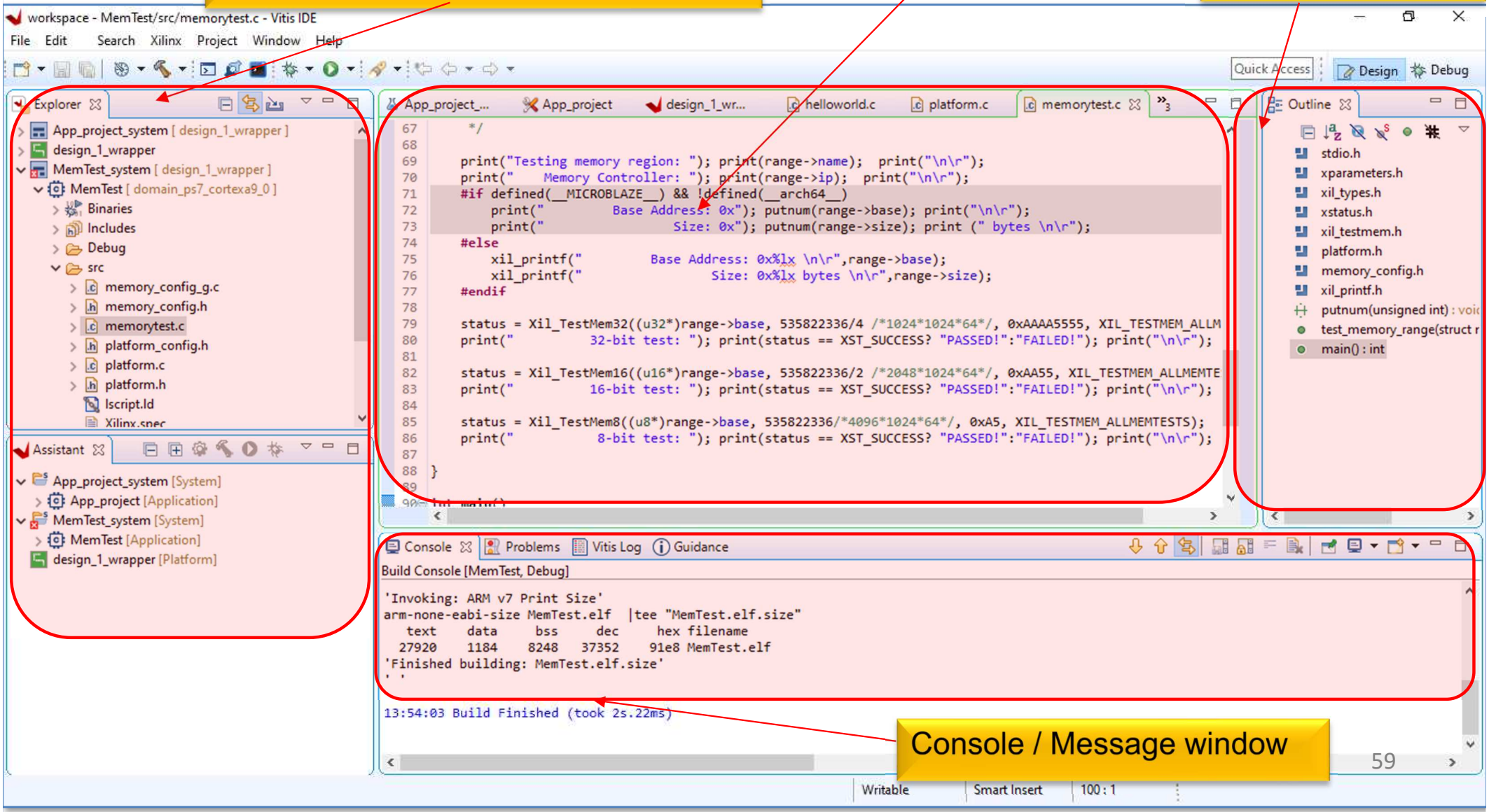
- Builder module:
  - Compiling and linking source files (linker script: **.LD)**
  - Generating compilation settings (option) for embedded SW application: select between Debug, Release and Profile configurations
  - These settings can be changed, refined later (custom build)
  - Compilation types: Standard Make, Managed Make (**makefile.**)
- Run module:
  - Application run (Run /+ Profile) vs. Debug
  - Target Connection settings
  - Executable/Run file: **.ELF + Executable Load File**
- Debug module:
  - GNU debugger (gdb), loading applicaiton, and HW debugging
  - Information about the status of debugging (Debug view)
- Search module/Help module:
  - Help in application development

VITIS SDK GUI

Project Explorer (3 level):
- HW Platform (design_1_wrapper)
- BSP: Board Support Package
- Application project

Source editor

Debug window



Console / Message window

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

THANK YOU FOR YOUR KIND ATTENTION!

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Strukturális és Beruházási Alapok

BEFEKTETÉS A JÖVŐBE