芯来科技
NUCLEI

Dual-Mode Configurable RISC-V Processor IP

芯来科技
Nuclei System Technology

芯来科技
NUCLEI

Dual-Mode Configurable

Configurable feature is to meet different application scenarios:

- Real-Time System Mode:
  - ✓ Instruction and Data Local Memory (ILM, DLM)
  - ✓ Physical Memory Protection (PMP)
  - ✓ Fast Interrupt Handling and Rich Interrupt Features (ECLIC)
- Application System Mode:
  - ✓ Instruction and Data Cache (I-Cache, D-Cache)
  - ✓ Memory Management Unit (MMU)
  - ✓ Private Timer
  - ✓ Platform Level Interrupt Controller (PLIC)
  - ✓ High Speed System Bus (AXI)

Dual-Mode Programmable

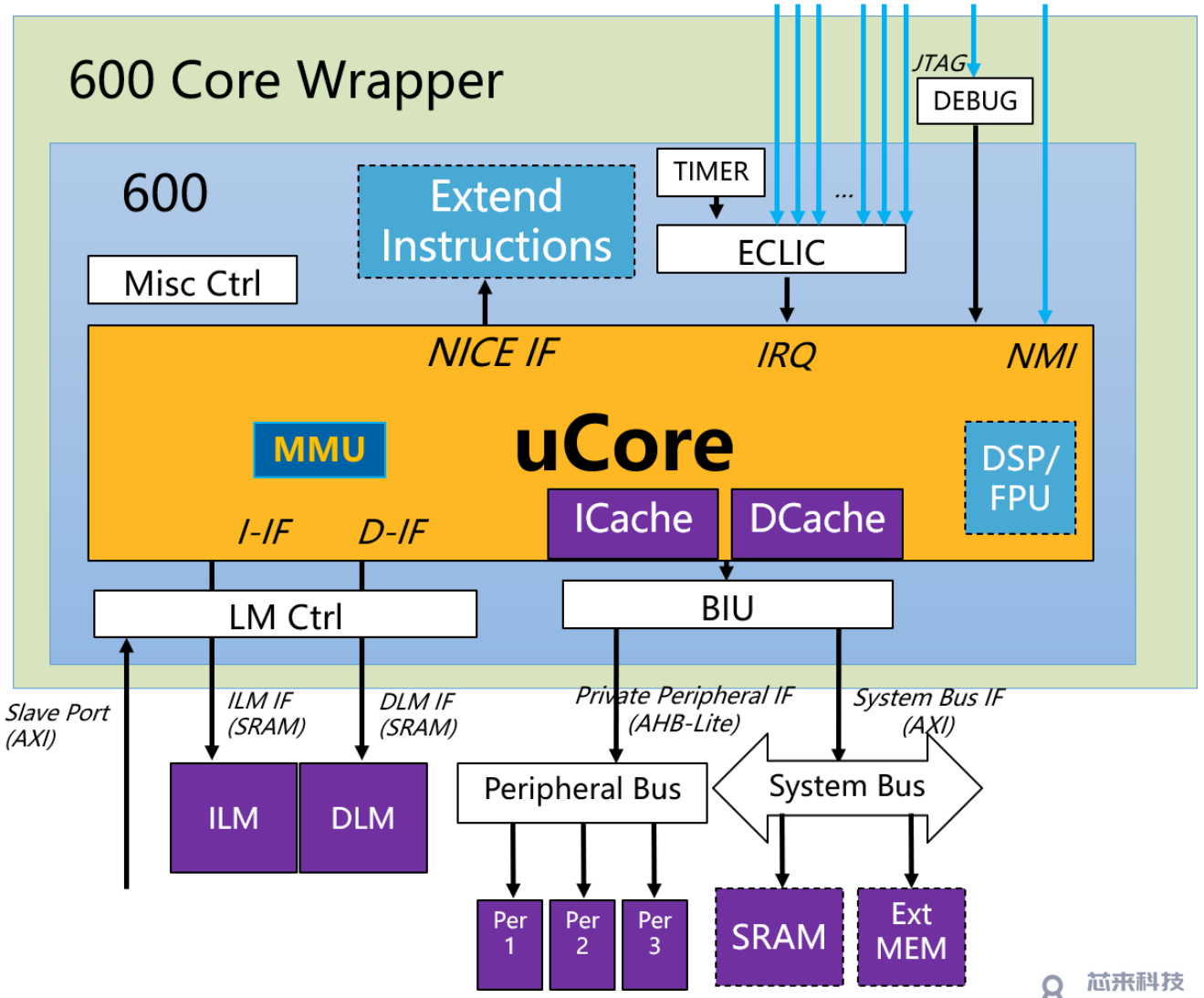
So here Dual-Mode configurable can be:

- Either Real-Time mode OR Application mode related features are configured
- Both the Real-Time mode AND Application mode related features are configured
  - ✓ Programmable: After Reset, SW can Enable/Disable features for the required MODE

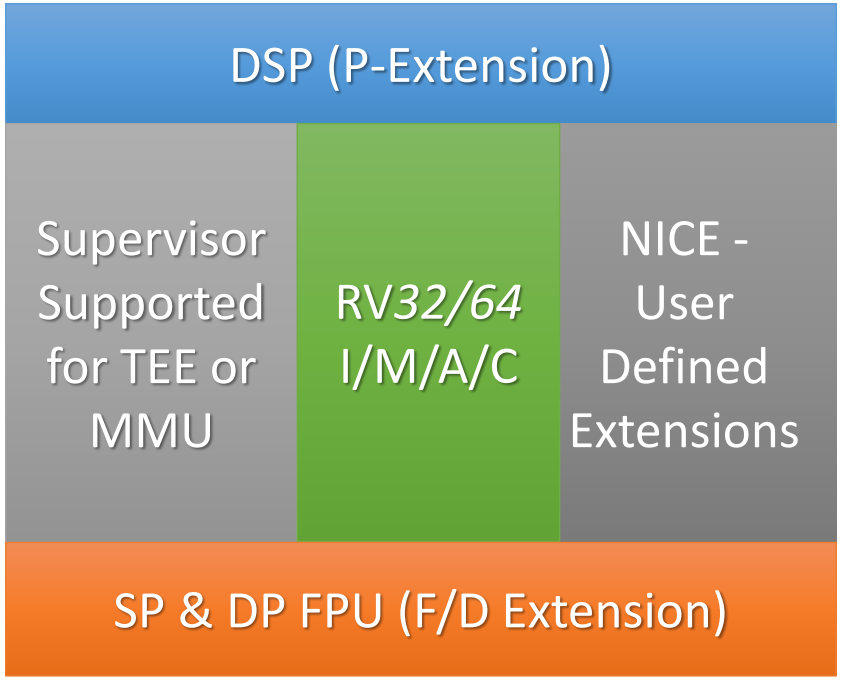
Nuclei 600 Series Processor

600 Series is configurable processor to meet different application scenarios.

- RISC-V **RV32/64-I/M/A/C/F/D/P** ISA supported
- **5-7 pipeline** stages
- Configurable **ILM** (Instruction Local Memory) & **DLM0/DLM1** (Data Local Memory) with **ECC**
- Configurable **I-Cache** & **D-Cache** with **ECC**
- **64-bit AXI** system bus, configurable 64-bit AXI slave port
- Besides **Machine mode & User mode, Supervisor mode** is suppored for **MMU** (RISC-V SV39 Mode)
- Configurable **NICE** interface for **user-defined extesnions**
- Configurable **ECLIC** (enhanced core level interrupt controller) or **PLIC** (platform level interrupt controller)
- 4-wire JTAG debug ports supported



600 Configurable ISA

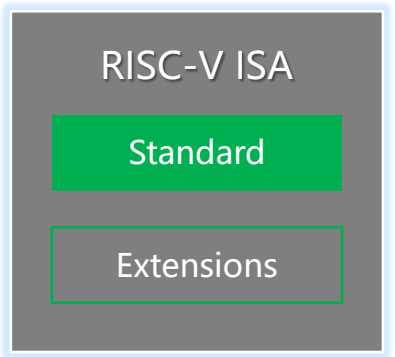


- Configurable & Extensible
- FPU options with both Single Precision & Double Precision
- DSP option with SIMD， partical SIMD, 64bit, Non-SIMD instructions
- Supervisor mode supported for Hardware-Software Co-design Penglai TEE in **N/NX** class cores
- Supervisor mode supported for MMU in **UX** class cores
- Instruction & Data closely coupled lcoal memories
- NICE interface for user-defined instruction extensions

NICE (Nuclei Instruction Co-unit Extension)

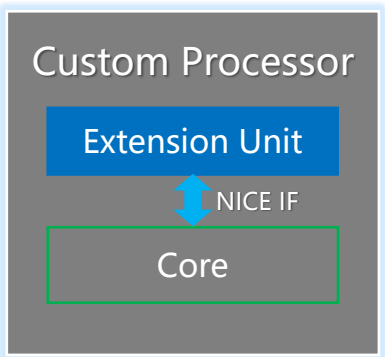
NICE allows customers to add user-defined instructions to customize their processor implementation, also including the extension of tightly coupled register and memory access instructions.

1. Define user-define instructions



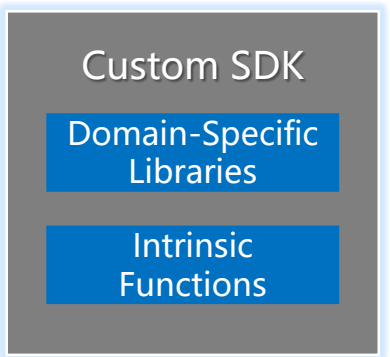
- Identify user-defined instructions by program profiling
- Define extension instructions into RISC-V reserved ISA space

2. Implement Extension Unit



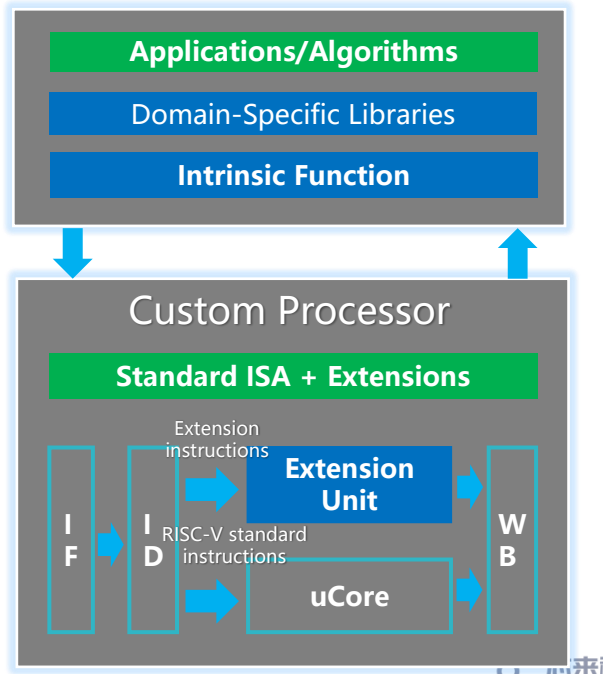
- Implement the application-specific co-unit following the NICE interface

3. Develop Domain Specific Lib/Function



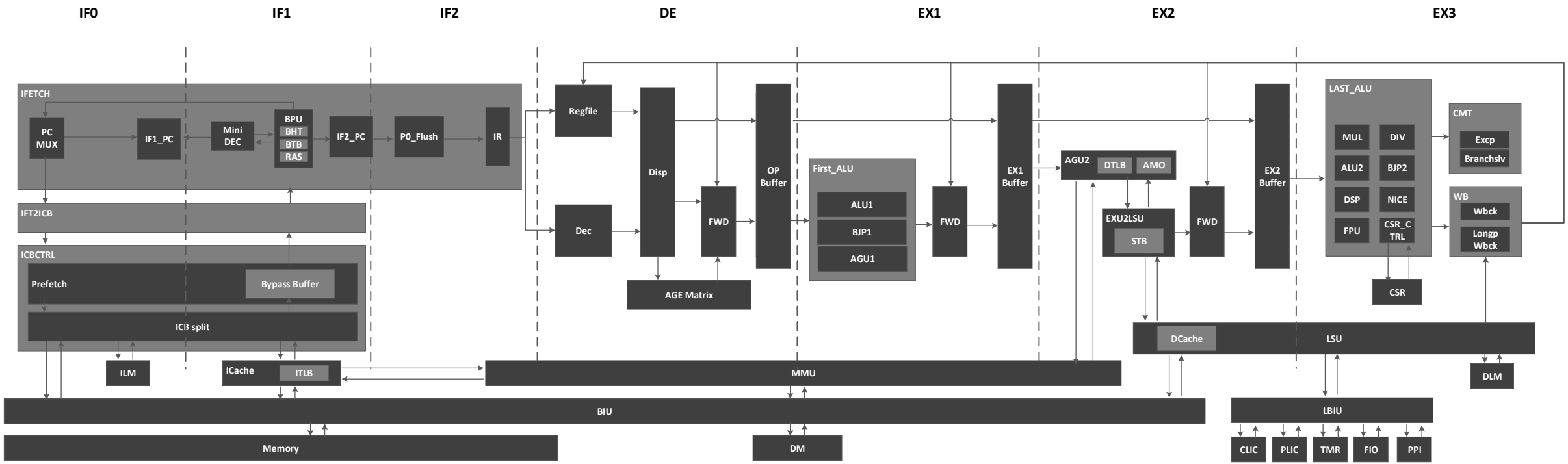
- No requirement for tool-chain update
- Using Intrinsic Function or encapsulated libraries

4. Accelerate Domain Specific Applications



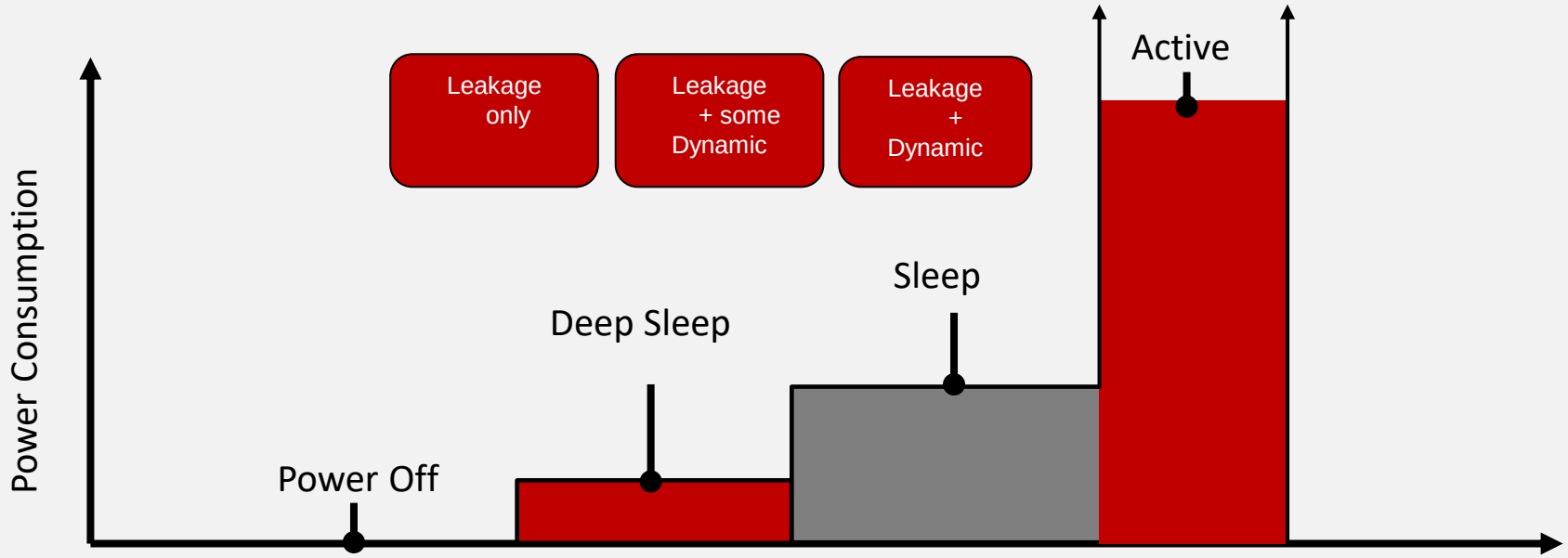
Low-Power Micro-Architecture Design

- 5-7 Pipeline Stages
- Various Low-Power Design (Clocking gating, Logic gating, etc.)



Sleep Modes

- Two Sleep Modes
  - Sleep mode & deep sleep mode, controlled by SoC MPU
- Entering Sleep Mode
  - WFI (wait for interrupt)
  - WFE (wait for event)
- Wake Up
  - NMI
  - Interrupt
  - Event
  - Debug Request



Memory Resources

- **ILM (Instruction Local Memory)**
  - Configurable an independent SRAM interface & address space
- **DLM (Data Local Memory)**
  - Configurable an independent SRAM interface & address space

Configuration	Comment
600_CFG_HAS_ILM	■This Macro configures to have ILM.
600_CFG_ILM_BASE_ADDR	■This Macro to configure the base address of the ILM.
600_CFG_ILM_ADDR_WIDTH	■This Macro to configure the address space of ILM.
600_CFG_HAS_DLM	■This Macro configures to have DLM.
600_CFG_DLM_BASE_ADDR	■This Macro to configure the base address of the DLM.
600_CFG_DLM_ADDR_WIDTH	■This Macro to configure the address space of DLM.
600_CFG_HAS_LM_SLAVE	■This Macro configures to have slave port for ILM/DLM access
600_CFG_HAS_LM_ECC	■This Macro configures to have ECC protections for ILM and DLM

- **Instruction Cache**
  - n-way associative, 4KB/way, n is configurable
  - Cache line size is 32 Byte
- **Data Cache**
  - 2-way associative, Cache line size is 32 Byte
  - Cache Size is configurable

Configuration	Comment
600_CFG_HAS_ICACHE	■This Macro configures to have I-Cache
600_CFG_ICACHE_WAY	■This Macro to configure n-way associate, n=2,4,8
600_CFG_HAS_DCACHE	■This Macro configures to have D-Cache
600_CFG_DCACHE_ADDR_WIDTH	■This Macro to configure the D-Cache size
600_CFG_HAS_CACHE_ECC	■This Macro configures to have ECC protections for I-Cache and D-Cache
600_CFG _DEVICE_REGIONn_BASE	■This Macro to configure the base address of Device Region n, n=0~7
600_CFG _DEVICE_REGIONn_MASK	■This Macro to configure the MASK value of Device Region n, n=0~7
600_CFG _NONCACHEABLE_REGIONn_BASE	■This Macro to configure the base address of Non-Cacheable Region n, n=0~7
600_CFG _ NONCACHEABLE _REGIONn_MASK	■This Macro to configure the MASK value of Non-Cacheable Region n, n=0~7

ECC on SRAMs

- **ECC Mechanism: SECDED (Single Error Correction, Double Error Detection)**
- **ECC protection granularity:**
  - ILM and I-Cache Data-Ram: 64-bit
  - DLM and D-Cache Data-Ram: 32-bit
  - I/D-Cache Tag-Ram and TLB: their Actual Size
- **ECC Full Write and Partial Write**
  - Full Write: data and corresponding ECC code will be updated simultaneously, high efficiency
  - Partial Write: Read-Modify-Write sequency will be triggered when 8/16-bit writes, less efficiency
- **ECC Error Injection**
  - mecc_code CSR is implemented for ECC error injection
  - Can be configured to do ECC error injection on ILM, DLM, I-Cache/D-Cache/TLB Tag-Ram or Data-Ram
- **ECC Lock**
  - ECC related CSRs cannot be modified after ECC is locked unless Reset, for Security

CCM (Cache Control and Maintenance)

- **CCM is defined for SW to Control and Maintenance the internal I-Cache and D-Cache**
- **CCM Types:**
  - by-**ADDR** and by-**ALL**
  - **I-Cache**: INVAL, INVAL_ALL, LOCK and UNLOCK
  - **D-Cache**: INVAL, FLUSH, FLUSH&INVAL, INVAL_ALL, FLUSH_ALL, FLUSH&INVAL_ALL, LOCK and UNLOCK
- **M/S/U mode has its own CCM operations**
  - **S/U** can execute CCM operations without needing switching privilege mode, but need to pass permission checking
  - 'Illegal instruction exception' will be triggered when lower privilege mode operates on higher privilege mode CCM CSRs
  - INVAL will be upgrade to be FLUSH&INVAL in U-mode for **Security**
- **CCM operations can still work on the Disabled cache**

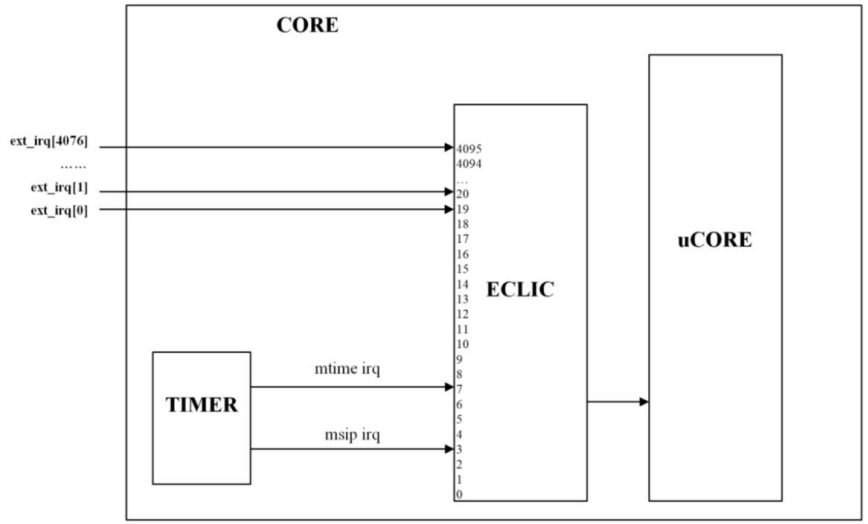
Type	Operation	Codes	Description
I-Cache operation	INVAL	5b01_000	When Cache hit, Unlock and Invalid the specific cacheline; Ignored when cache miss.
	LOCK	5b01_011	When cache hit, Lock the specific cacheline; When cache miss, Refill the specific cacheline then Lock it. Check ccm_data to check the Lock succeed or not.
	UNLOCK	5b01_100	When cache hit, Unlock the specific cacheline; Ignored when cache miss.
	INVAL_ALL	5b01_101	Unlock and Invalid ALL the cacheline.
D-Cache operation	INVAL	5b00_000	When cache hit, Unlock and Invalid the specific cacheline; Ignored when cache miss.
	WB	5b 00_001	When cache hit and Dirty, Flush the specific cacheline; When cache miss or hit but not dirty, ignored. Lock bit is not affected.
	WBINVAL	5b 00_010	When cache hit, Unlock and Flush and Invalid the specific cacheline; Ignored when cache miss.
	LOCK	5b 00_011	When cache hit, Lock the specific cacheline; When cache miss, Refill the specific cacheline then Lock it. Check ccm_data to check the Lock succeed or not.
	UNLOCK	5b 00_100	When cache hit, Unlock the specific cacheline; Ignored when cache miss.
	INVAL_ALL	5b 10_111	Unlock and Invalid ALL the cacheline.
	WB_ALL	5b 00_111	Flush ALL the Valid and Dirty cachelines; Lock bit is not affected.
	WBINVAL_ALL	5b 00_110	Unlock and Flush and Invalid ALL the Valid and Dirty cachelines.

Bus Interfaces

- **System Bus Interface** - 64bit AXI with integer clock ratios
- **ILM Bus Interface** - 64bit (configurable) SRAM interface, for accessing private instruction local memory
- **DLM Bus Interfaces** – 2 32bit (configurable) SRAM interfaces, for accessing private data local memory (DLM0/DLM1)
- **Private Peripheral Interface (PPI)** - 32bit, AHB-Lite interface protocol for accessing private peripherals
- **Slave Port** – 64bit AXI interface for other masters to access ILM/DLM0/DLM1

Enhanced Core Local Interrupt Controller

- ECLIC (Enhanced Core Local Interrupt Controller)
  - Optimized based on the RISC-V standard CLIC for fast interrupt handling scheme, **compatible with CLIC**
  - **Private inside the core**
  - Enabled by setting the LSB bits of CSR register **mtvec** as CLIC/ECLIC mode
  - Configurable number of interrupt levels and priorities
  - Support interrupt preemptions based on interrupt levels
  - Support vectored interrupt processing mode for extremely fast interrupt response (6 cycles)
  - Support fast interrupts tail-chaining mechanism (non-vectored)



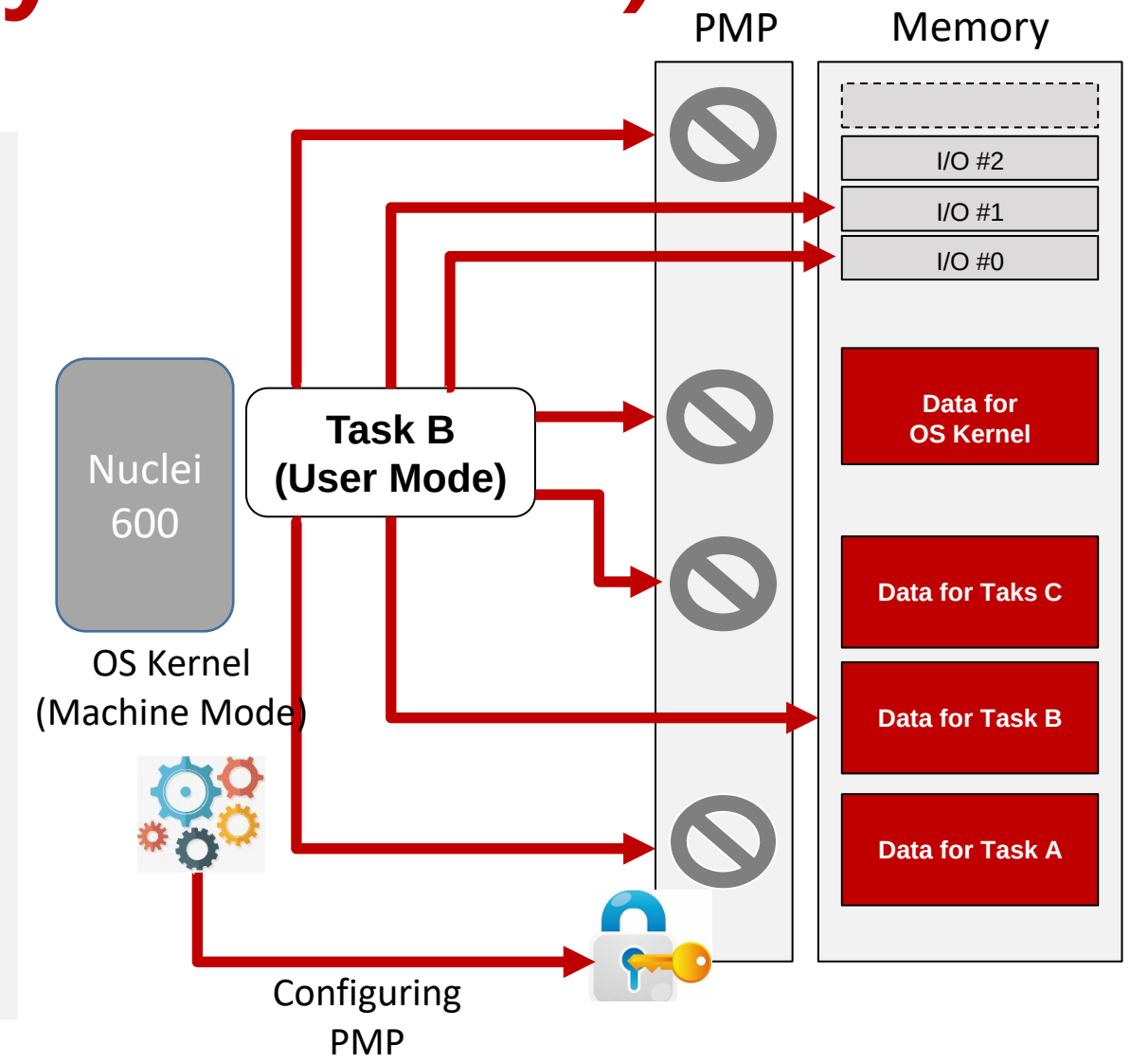
CSR Registers	Comment
mtvt	ECLIC Interrupt Vector Table Base Address
mnxti	Used to enable taking the next interrupt and return the entry address of the next interrupt handler.
mintstatus	Current Interrupt Levels
mnvec	Customized register used to indicate the NMI handler entry address
mmisc_ctl	Customized register controlling the selection of the NMI Handler Entry Address.
msavestatus	Customized register storing the value of mstatus.
mtvt2	Customized register used to indicate the common handler entry address of non-vectored interrupts.
jalmnxti	Customized register used to enable the ECLIC interrupt. The read operation of this register will take the next interrupt, return the entry address of next interrupt handler, and jump to the corresponding handler at the same time.
pushmcause	Customized register used to push the value of mcause into the stack memory.
pushmepc	Customized register used to push the value of mepc into the stack memory

Platform Level Interrupt Controller

- PLIC (Platform Level Interrupt Controller)
  - RISC-V standard, for Linux Capable or SMP applications
  - **Shared outside of the core**
  - Enabled by setting the LSB bits of CSR register **mtvec** as CLINT mode
  - Up to 1024 Interrupts supported: level or edged
  - Configurable number of interrupt priorities
  - Support interrupt to M-mode or S-mode
  - PLIC Control Registers:
    - ✓ Interrupt Enable
    - ✓ Interrupt Pending
    - ✓ Interrupt Priority
    - ✓ Interrupt Threshold
    - ✓ Interrupt Claim/Complete

PMP (Physical Memory Protection)

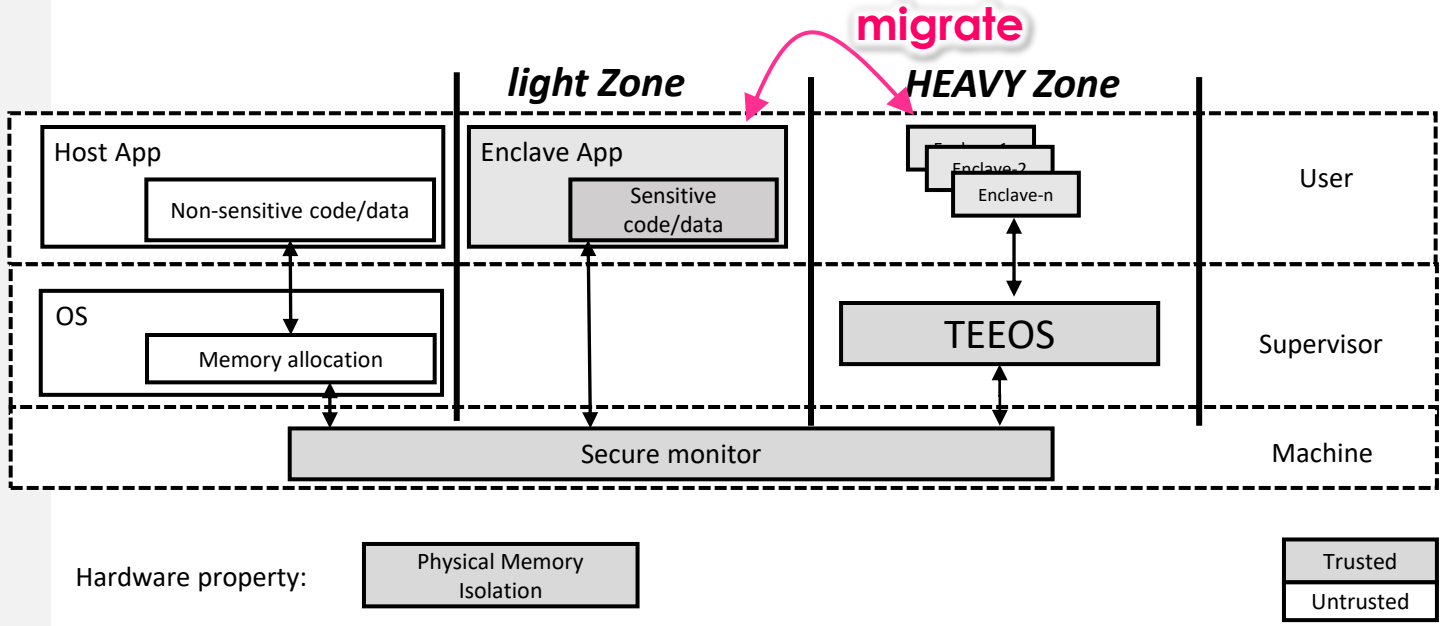
- Configurable PMP, providing per-hart machine-mode control registers to allow physical memory access privileges (read, write, execute) to be specified for each physical memory region.
  - **Configurable PMP entries, up to 16**
  - The granularity of PMP is **4KB**
  - TOR mode is not supported
  - PMP checks are applied to all accesses when the hart is running in S or U modes; And for loads and stores when the MPRV bit is set in the mstatus register and the MPP field in the mstatus register contains S or U



TEE (Trusted Execution Environment)

- RISC-V Privileged ISA based TEE Framework
- Smallest Trusted Code Base
  - RISC-V core (PMP/sPMP) + Verifiable security monitor (M-mode privilege) + TEEOS
- Secure Assurance
  - Strong isolation between enclave and other application or OS
  - Protect against a malicious or compromised OS
  - Secure boot and remote attestation for chain of trust
  - High performance and scalability

Penglai TEE Architecture



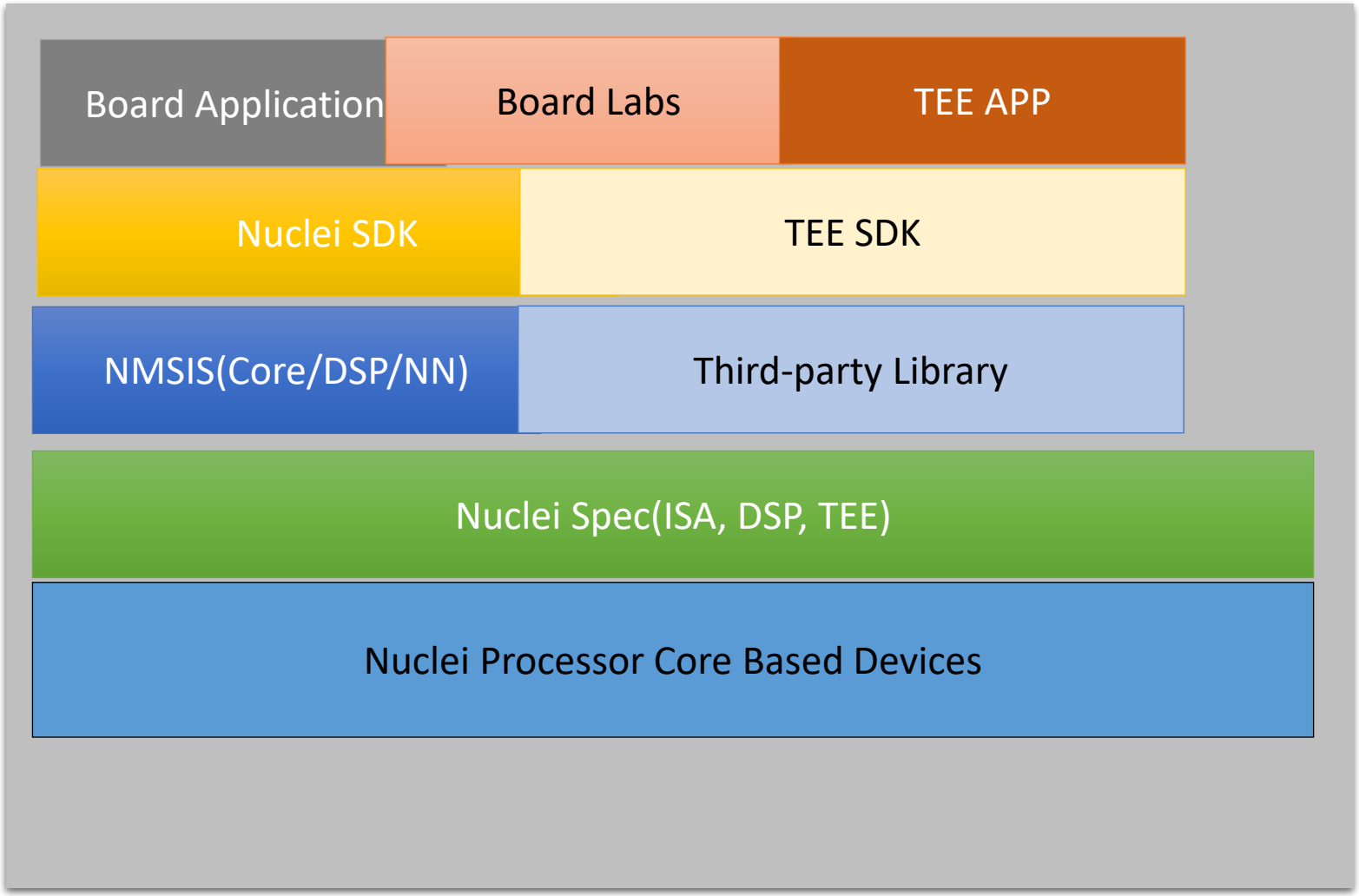
Penglai HEAVY.light architecture

- **light Zone:** A dedicated HW-isolated box for a single enclave
- **HEAVY Zone:** Multiple-Enclaves isolated through TEEOS

MMU (Memory Management Unit)

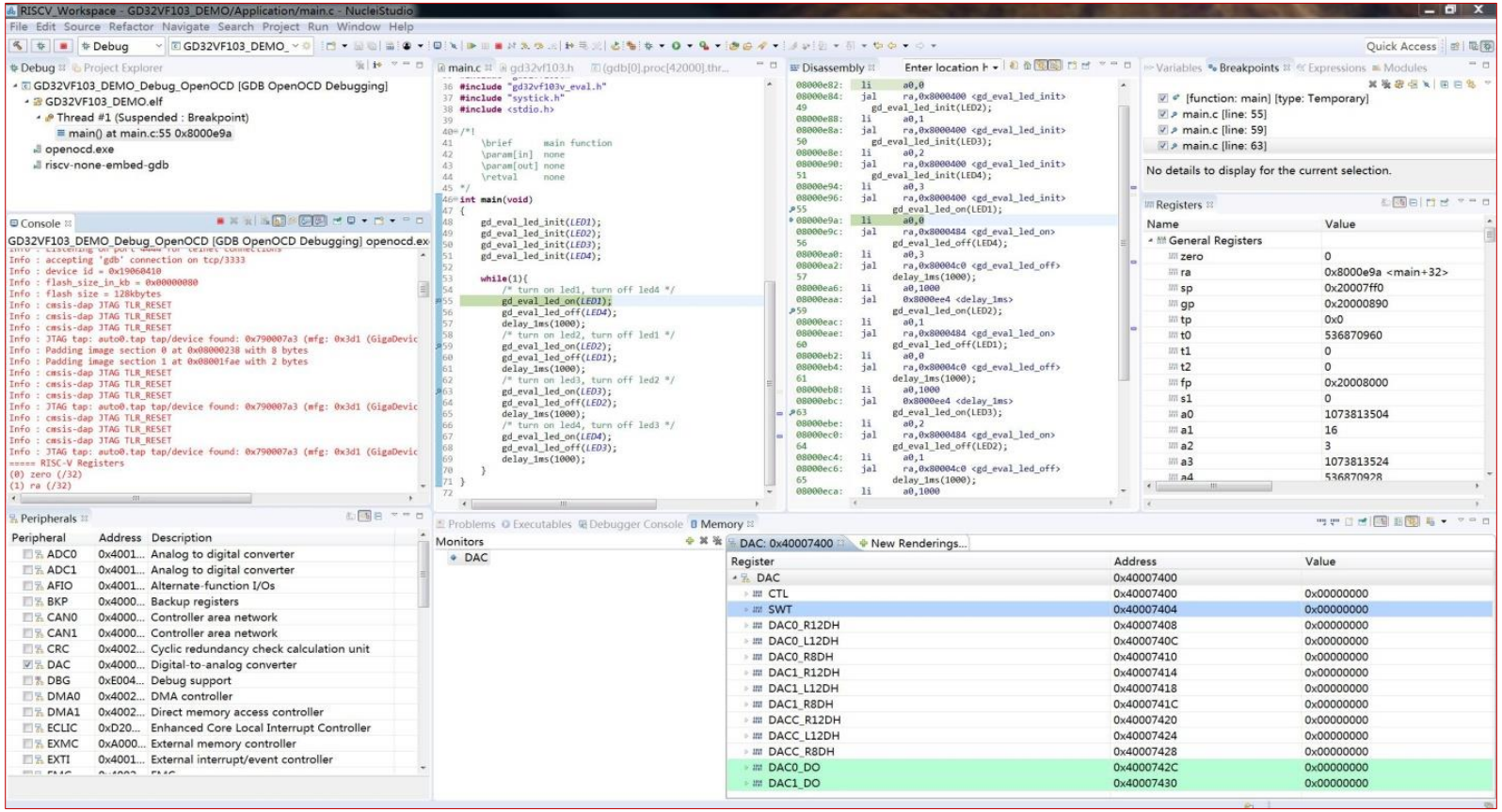
- MMU in **UX** class cores can enable Linux capable applications
- RISC-V RV39 mode:
  - ✓ Page based 39-bit virtual memory system, mapping to 56-bit physical memory space
  - ✓ Permission checking (e**X**ecutable, **W**riteable, **R**eadable)
- Two level TLBs (Translation Lookaside Buffer) in MMU to cache page tables for fast accessing:
  - ✓ Main TLB: can be configured as 32, 64 or 128 entries
  - ✓ Micro TLB (I-TLB, D-TLB): each has 8 entries
- MMU supports 4KB, 2MB and 1GB page types
- Hardware Translation Table Walk mechanism when TLB missing without software handling

Nuclei Software Development Platform



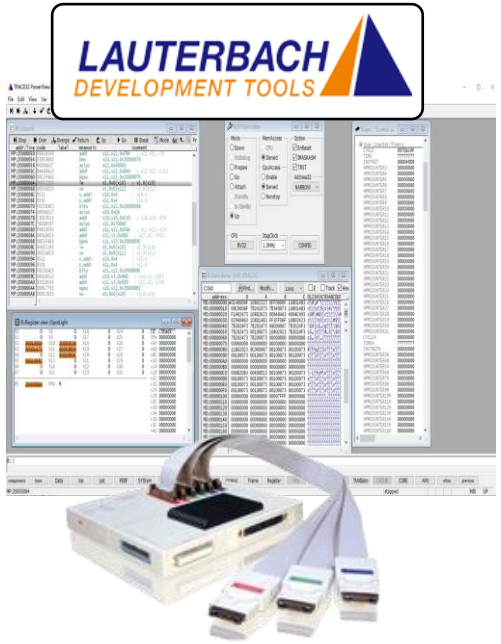
Nuclei Studio IDE

- Eclipse based
- Integrated GCC and OpenOCD
- Libre and free
- Portable executables, without installation
- Easy-to-use project template
- Integrated editor
- In system debugging
- In system programming
- Integrated serialport tool
- Real time register display

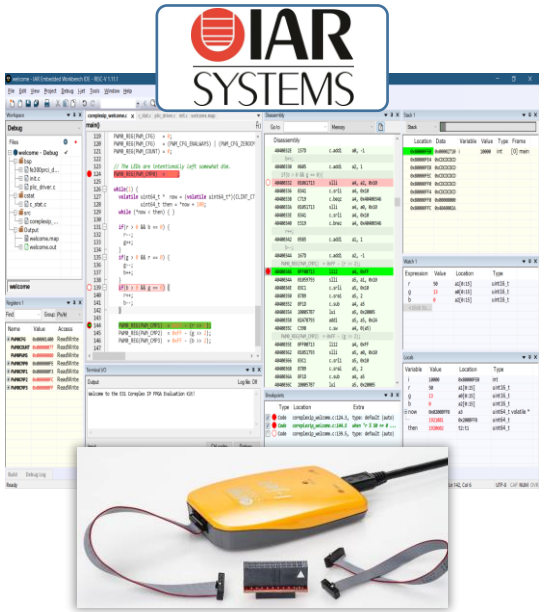


Supported 3rd Party Tools

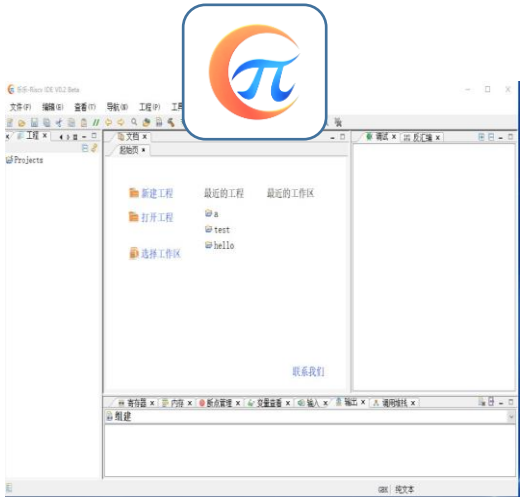
SEGGER



LAUTERBACH



IAR



COMPLIER

芯来科技
NUCLEI

THANK YOU

芯来科技
NUCLEI