

FPGA Development with Agilex™ 3

Practicing Nios® V and Quartus®!

Macnica, Inc.
AlTIMA Company



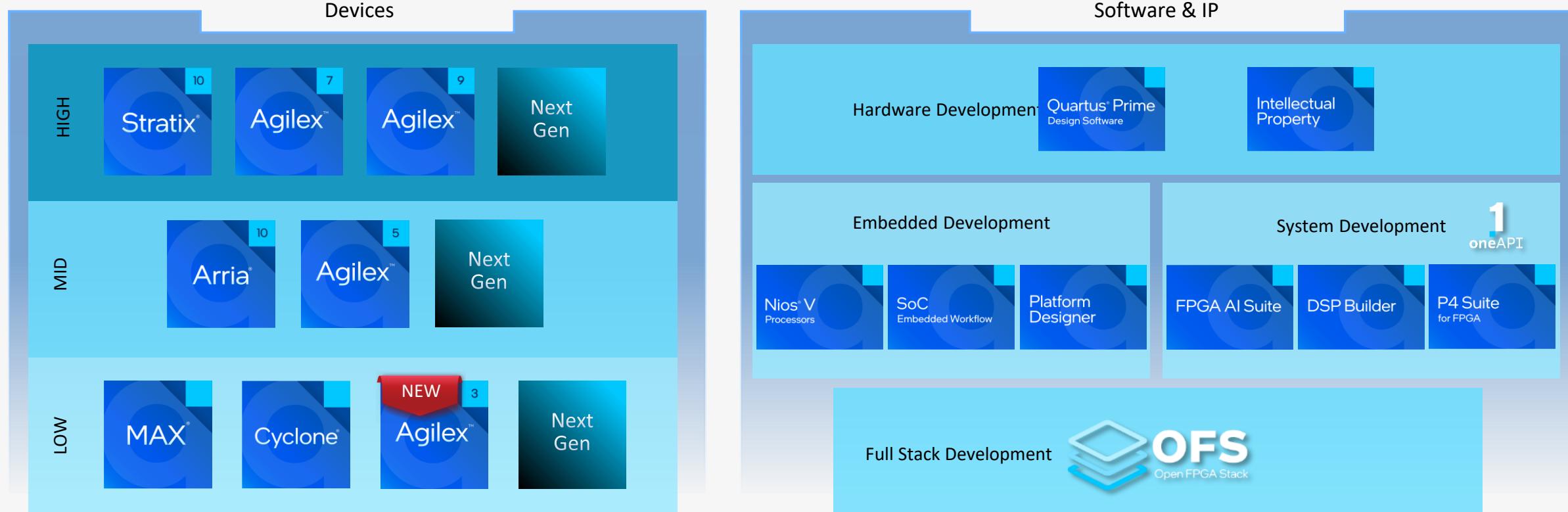
Agenda

1. Agilex™ 3 FPGA & SoC FPGA Overview
2. What is Nios® V?
3. Migrating from Nios® II
4. Nios® V Hardware Development
5. Exercise 1: Hardware Exercise
6. Nios® V Software Development
7. Nios® V Boot Method
8. Introducing Debugging Tools
9. Exercise 2: Software Exercise
10. What is the FPGA AI Suite?
11. Introduction of Materials
12. Summary

Agenda

1. Agilex™ 3 FPGA & SoC FPGA Overview
2. What is Nios® V?
3. Migrating from Nios® II
4. Nios® V Hardware Development
5. Exercise 1: Hardware Exercise
6. Nios® V Software Development
7. Nios® V Boot Method
8. Introducing Debugging Tools
9. Exercise 2: Software Exercise
10. What is the FPGA AI Suite?
11. Introduction of Materials
12. Summary

FPGA portfolio of speed, choice and simplicity



Partner Ecosystem

Agilex™ 3 FPGAs take cost-optimized design performance to the next level



- Enable performance in small devices
- Features
-



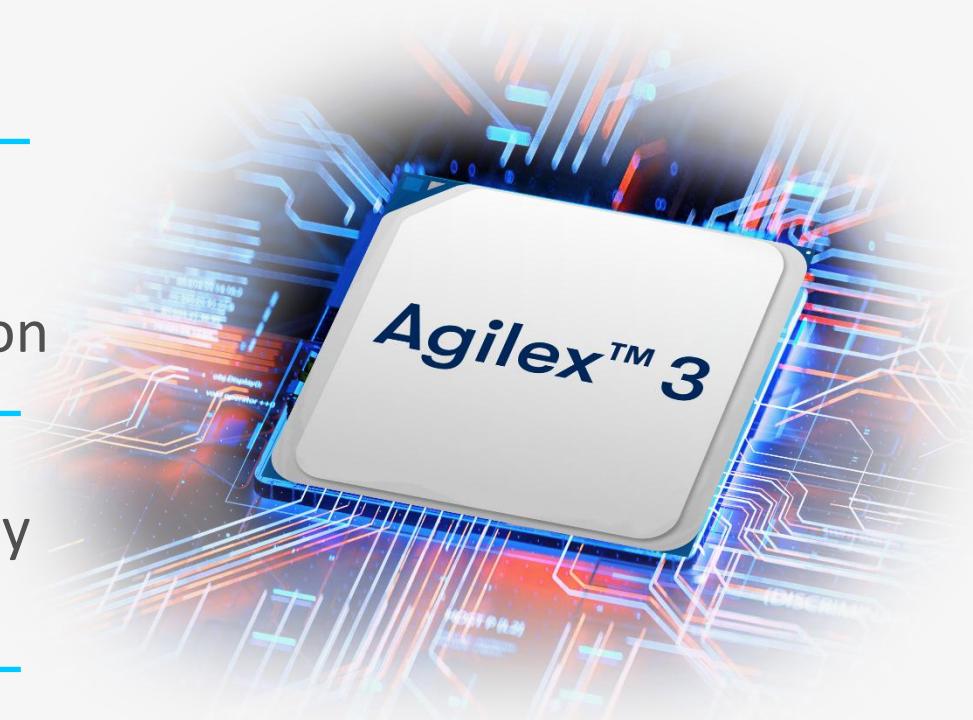
- Advanced functionality integration with hard IP and packaging for cost optimization
-



- Protect your IP from theft Leading security features
-



- Intel's diverse and resilient supply
Guarantees short- and long-term supply continuity



Source: Altera

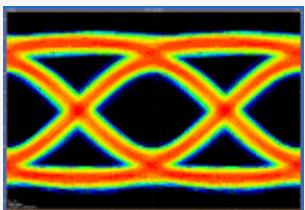
Agilex™ 3 FPGAs Technology Enablers for Power-Efficient Performance



- Intel leadership Technology
 - Intel 7 process technology with interconnect innovations
 - Intel VPBGA packaging technology increases IO density



- 2nd Generation Hyperflex™ FPGA Architecture
 - Delivers higher performance/watt than Cyclone® V FPGAs

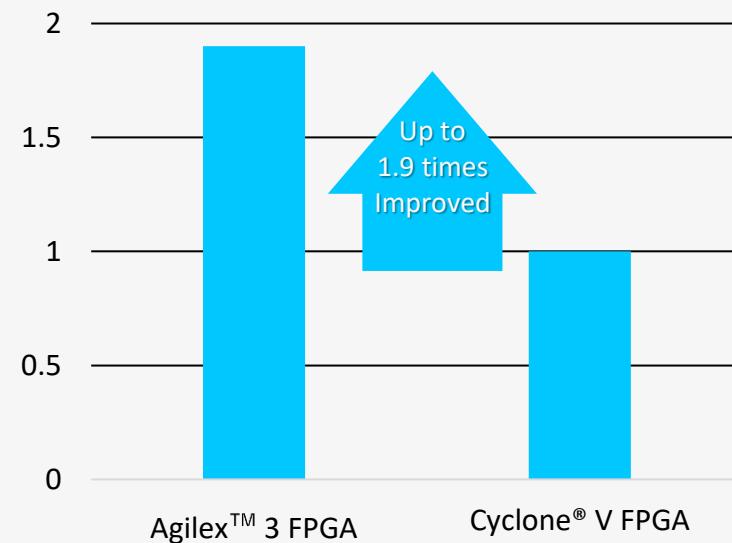


- High level of system integration
 - High-speed transceiver I/O, Ethernet hard IP, DDR controller hard IP , and optional Hard Processor System (HPS)

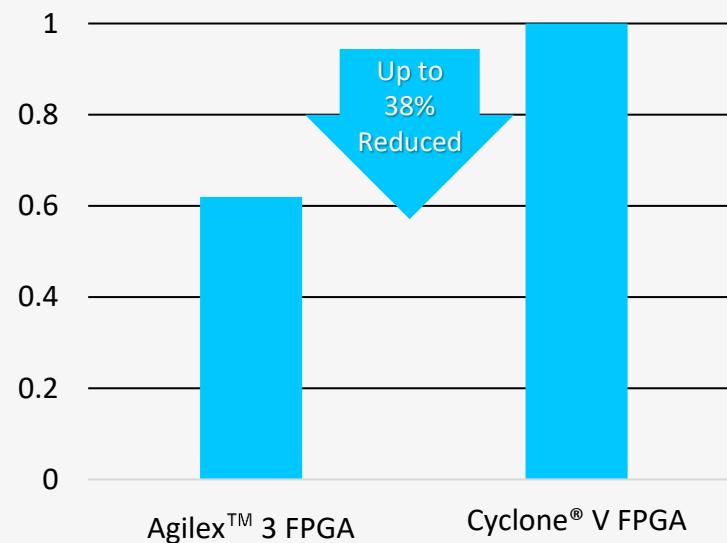
Agilex™ 3 FPGAs deliver better performance and lower power consumption

- Up to 1.9X higher fabric performance or up to 38% lower total power compared to Cyclone® V FPGA(1)

Agilex™ 3 FPGAs⁽²⁾ vs.
Cyclone® V FPGAs



Agilex™ 3 FPGAs⁽²⁾ vs.
Cyclone® V FPGAs



Notes:

1. Fabric performance comparison is done between Cyclone® V FPGA and Agilex™ 3 FPGA
2. The performance QoR (Quality of Results) Fmax (maximum frequency) represents the geomean derived from Altera's QoR design suite of 45 separate designs.
3. The total power comparison is done with Agilex™ 3 core power at 0.75V and Cyclone V® core power at 1.1V and both devices at 100% utilization running at 150 MHz.
4. Performance varies by use, configuration and other factors. Learn more at www.intel.com/PerformanceIndex. Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Agilex™ 3 Family Table

Product Lineup		A3C025	A3C050	A3C065	A3C100	A3C135	Specification code	HPS	EMIF, MIPI, PUF, SPDM authentication		
Logic	Specification Code	Y, Z	Y, Z	Y, Z	W, Y, Z	W, Y, Z	W	Y	Y		
	Logic Elements	25K	47K	65K	100K	135K	Y	N	Y		
Internal RAM	ALM	8,500	16,000	22,000	34,000	45,800	Z	N	N		
	M20K	65	123	169	262	353					
DSP	M20K(Mb)	1.28	2.4	3.3	5.1	6.9					
	Variable Precision DSP Block	34	65	88	138	184					
Transceiver	@ 12.5 Gbps	0	0	0	4	4					
	PCIe	0	0	0	3.0 x4	3.0 x4					
Hard IP	10GbE Hard IP	0	0	0	1	1					
	MIPI D-PHY @ 2.5Gbps	0	7	7	14	14					
GPIO	HVIO (1.0V- 3.3V)	160	160	160	160	160					
	HSIO (1.0V -1.3V)	96	96	96	192	192					
EMIF	external memory interface	-	LPDDR4	LPDDR4	LPDDR4	LPDDR4					
PLL	IO/Fabric	2/5	2/5	2/5	4/8	4/8					
HPS	Hard processor subsystem	N/A	N/A	N/A	2x A55 up to 800 MHz						
Security features	Secure Device Manager (SDM)	SHA -384, ECDSA 256/384 Bitstream Authentication, AES -256, Side Channel Attack Resistance, Cryptographic Services, PUF, SPDM Authentication, Physical Tamper Protection support									
Package code (Package size, ball pitch, and array pattern)		HVIO/HSIO/HPSIO/XCVR 									
M12A (12x12mm, 0.5 mm, Standard BGA)		160/72/0/0	160/72/0/0	160/72/0/0	40/192/48/4	40/192/48/4					
M16A (16x16mm, 0.5 mm, Standard BGA)		160/48/0/0	160/48/0/0	160/48/0/0	160/48/0/0	160/48/0/0					
B18A ² (18x18mm, Variable, VPBGA ¹)		160/96/0/0	160/96/0/0	160/96/0/0	200/144/48/4	200/144/48/4					
B18B (18x18mm, Variable, VPBGA ¹)											
B23A (23x23mm, Varaiable, VPBGA ¹)											

Definitions:

- HVIO: High-speed IO
- HPS: High-voltage IO
- HPSIO: HPS IO available with HPS
- VPBGA: Variable Pitch Ball Grid Array
- XCVR: transceiver

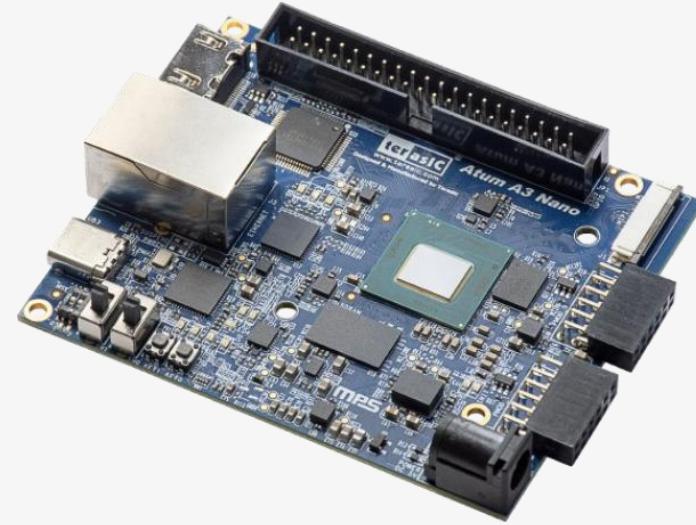
Notes:

- 1) Package ball pitch is variable, please contact to your sales representative for more information.
- 2) B18A does not support specification W
- 3) A3C025 does not support EMIF

Evaluation board



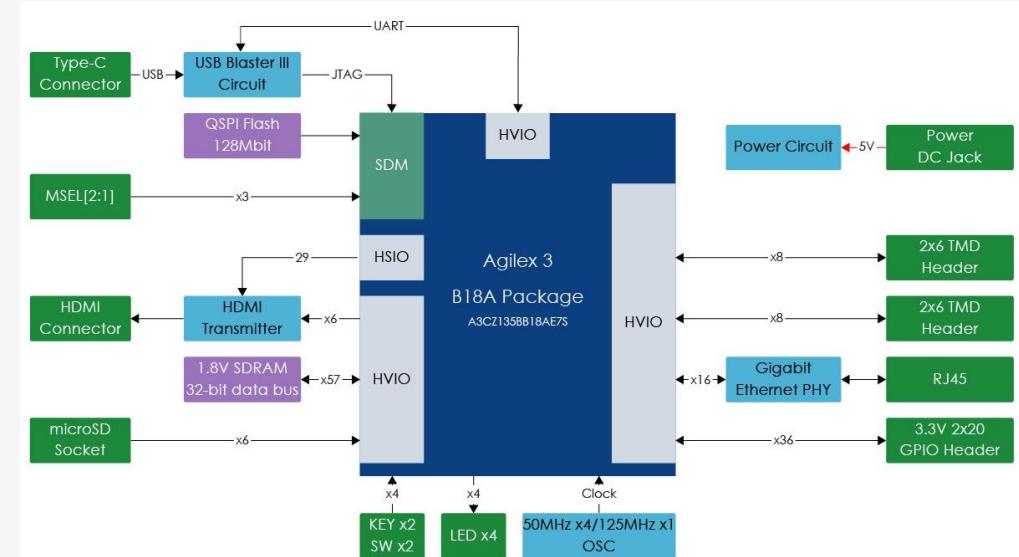
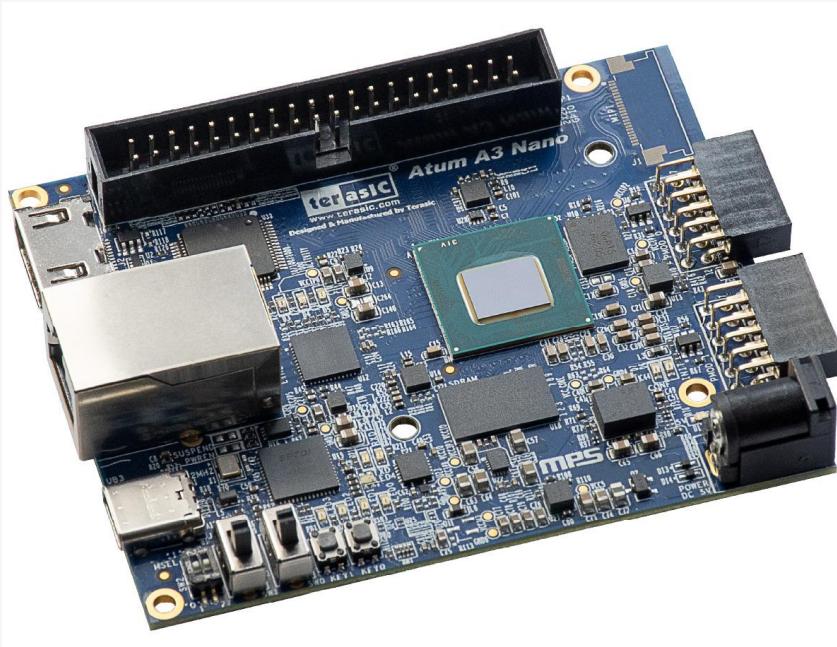
[Agilex™ 3 FPGA C-Series
Development Kit](#)



[Terasic Atum A3 Nano
Development Kit](#)

Evaluation board use today: Terasic Atum A3 Nano

- FPGA
 - Agilex™ 3
 - A3CZ135BB18AE7S
- Memory Device
 - 64MB SDRAM, x32 bits data bus
 - MicroSD Card Socket
- Communication
 - Gigabit Ethernet
 - UART
- Switches/Buttons/LEDs
 - 4 LEDs
 - 2 Slide Switches
 - 2 Push Buttons



[Terasic - All FPGA Boards - Agilex 3 - Atum A3 Nano](#)

Agenda

1. Agilex™ 3 FPGA & SoC FPGA Overview
2. What is Nios® V?
3. Migrating from Nios® II
4. Nios® V Hardware Development
5. Exercise 1: Hardware Exercise
6. Nios® V Software Development
7. Nios® V Boot Method
8. Introducing Debugging Tools
9. Exercise 2: Software Exercise
10. What is the FPGA AI Suite?
11. Introduction of Materials
12. Summary

Next-generation Nios® V processors for Altera® FPGAs

RISC-V architecture

Open-spec RISC-V architecture

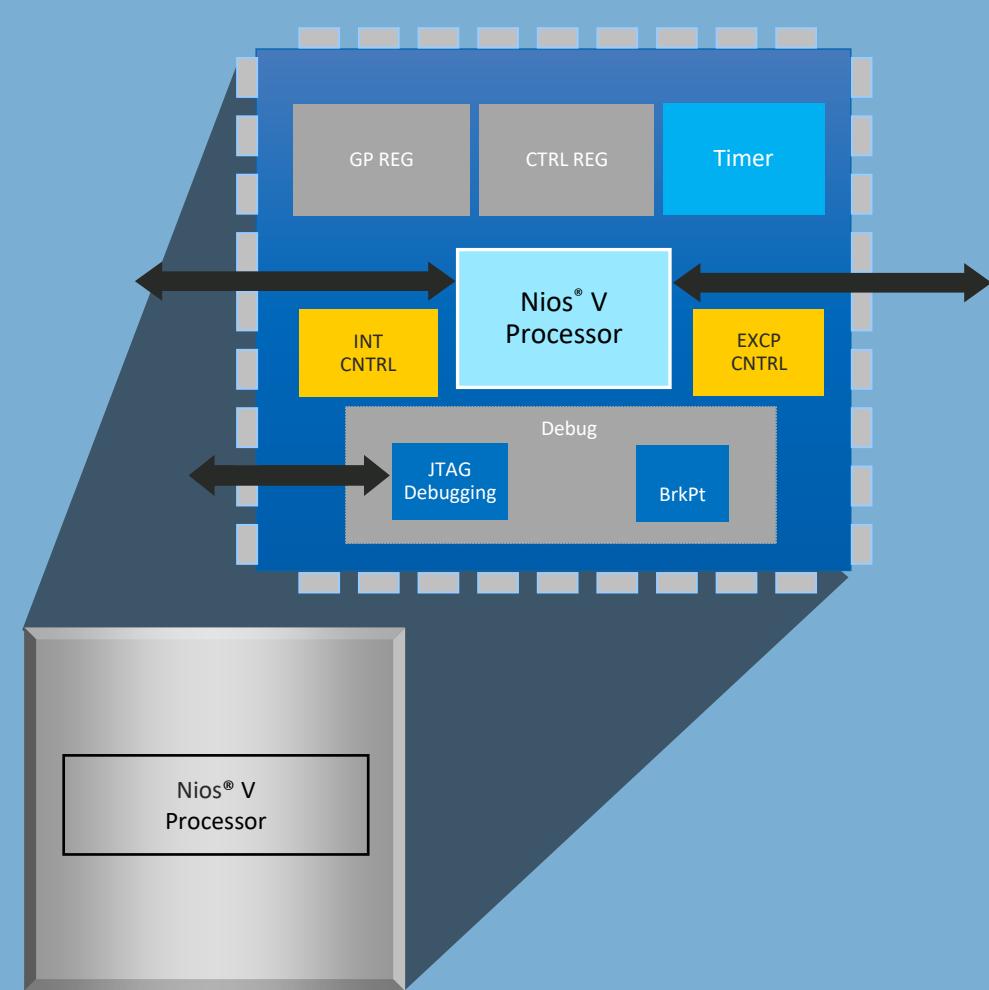
Rich ecosystem

Latest IDEs, compilers, debuggers,
Operating System (OS), etc.

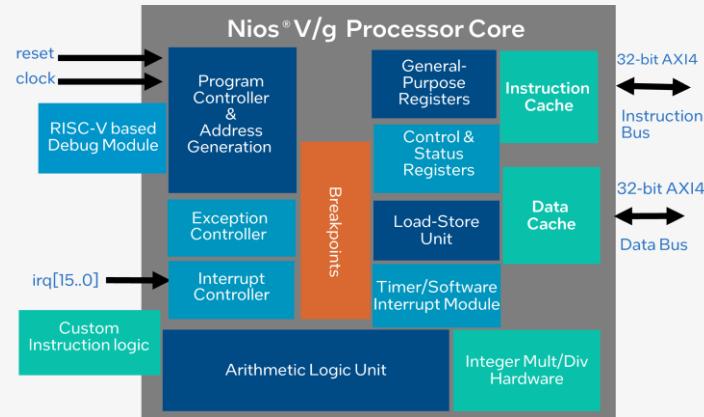
Available now

Quartus® Prime Development Software
Pro Edition version 21.3 or later

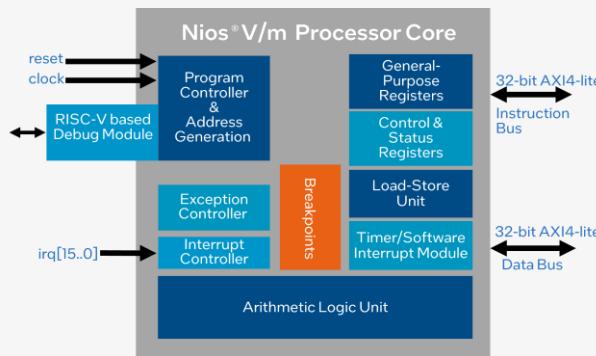
Quartus® Prime Development Software
Standard Edition version 22.1 or later



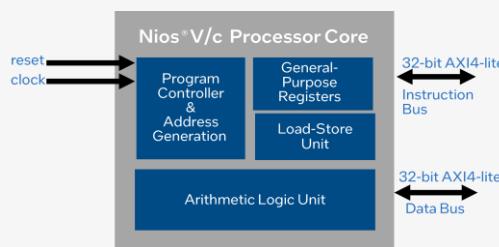
Nios® V: Choose from Three Types Based on Your Application Needs



- Nios® V/g general-purpose processor
 - Highest performance processor



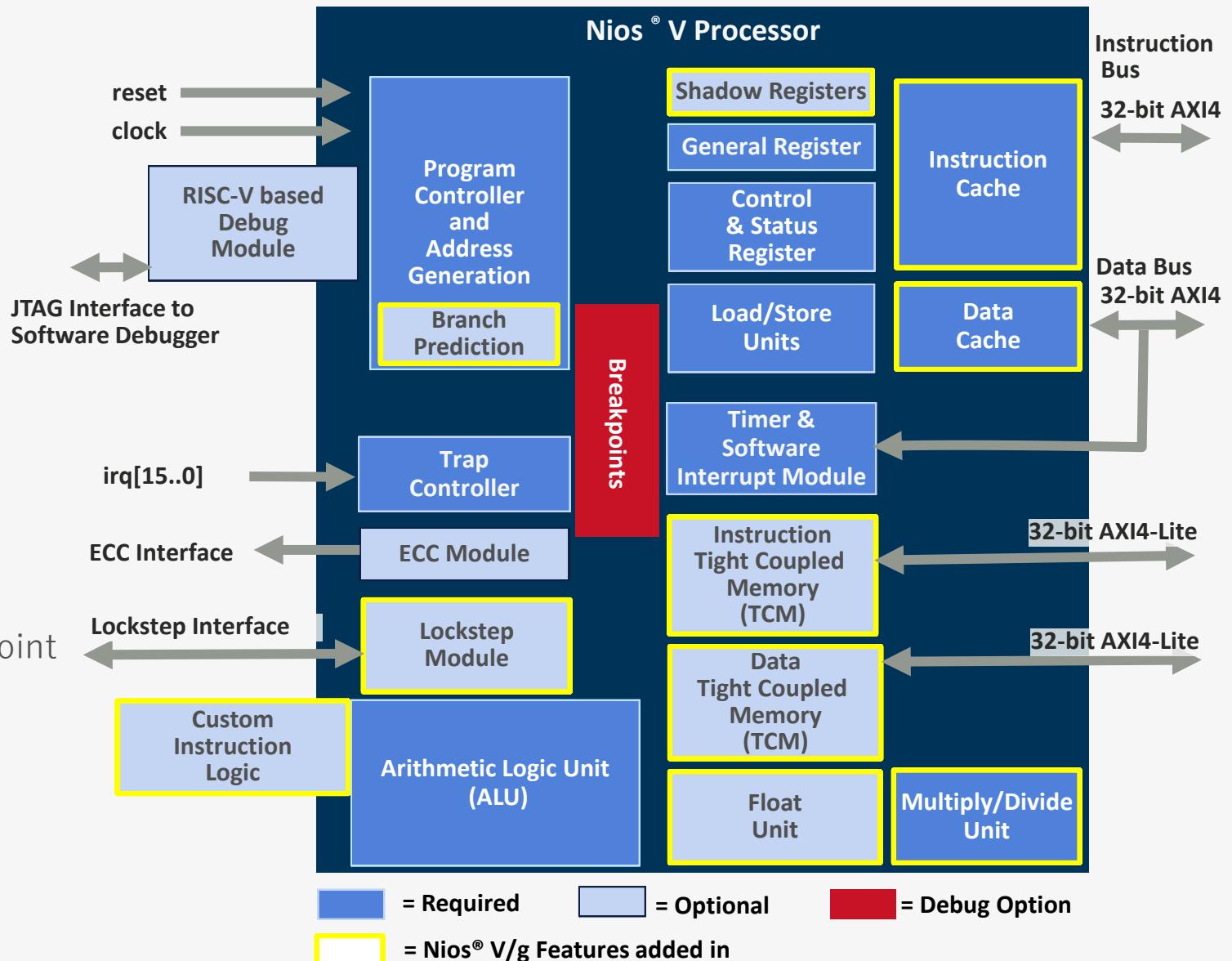
- Nios® V/m microcontroller
 - Balanced processor between size and performance



- Nios® V/c compact microcontroller
 - Ultra-compact controller for size-constrained applications

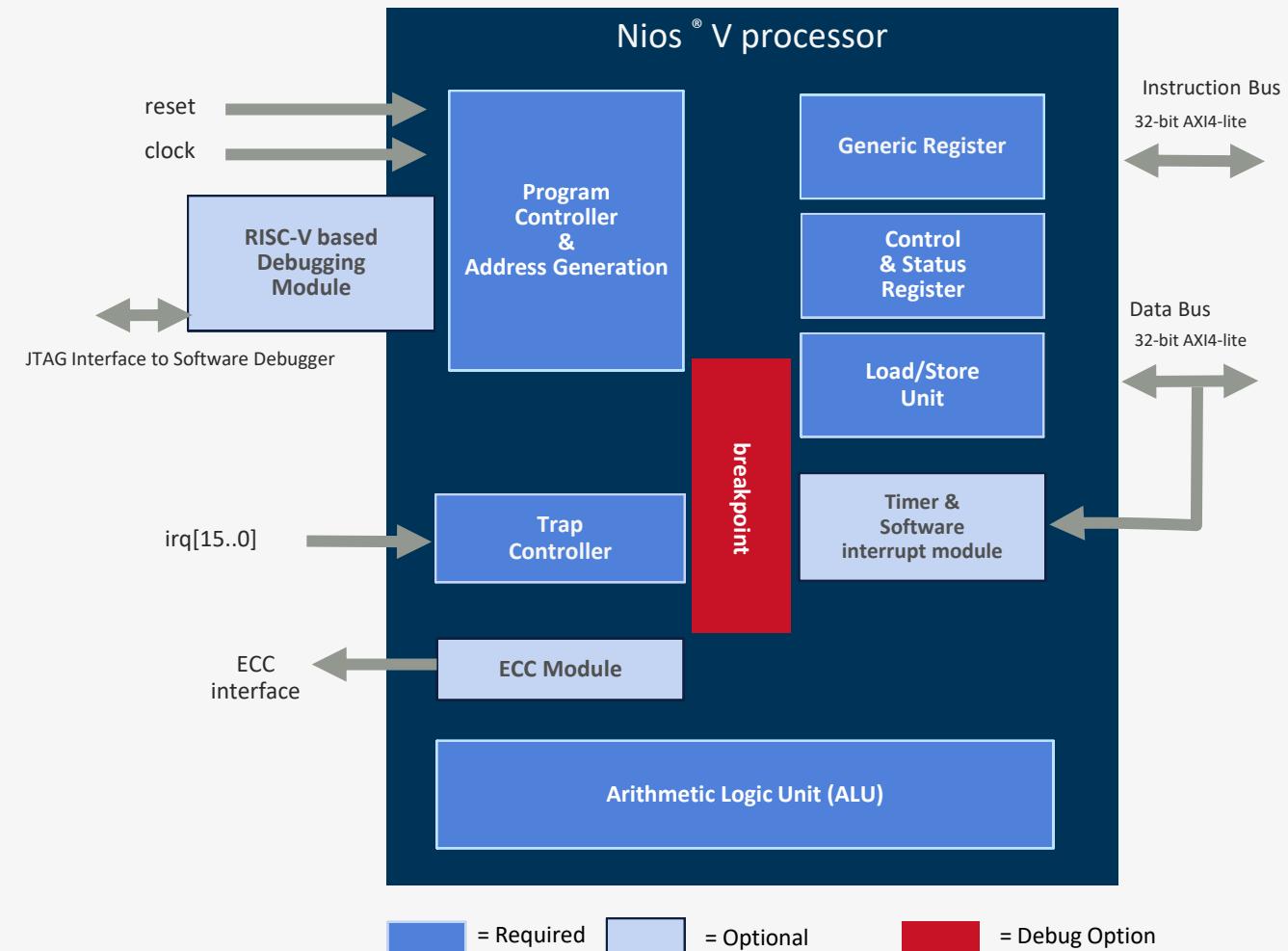
Nios® V/g Processor Core

- General purpose processor
- RV32IM (F) Zicsr Zicbom RISC-V instruction set
- Supports all FPGAs
 - 23.1pro/23.1std and later
- Differences from Nios® V/m
 - Cache/Custom Instructions/Integer Multiplication/Division Unit/Floating-Point Unit (FPU)/Tightly Coupled Memory (TCM)/Branch Prediction/Lockstep



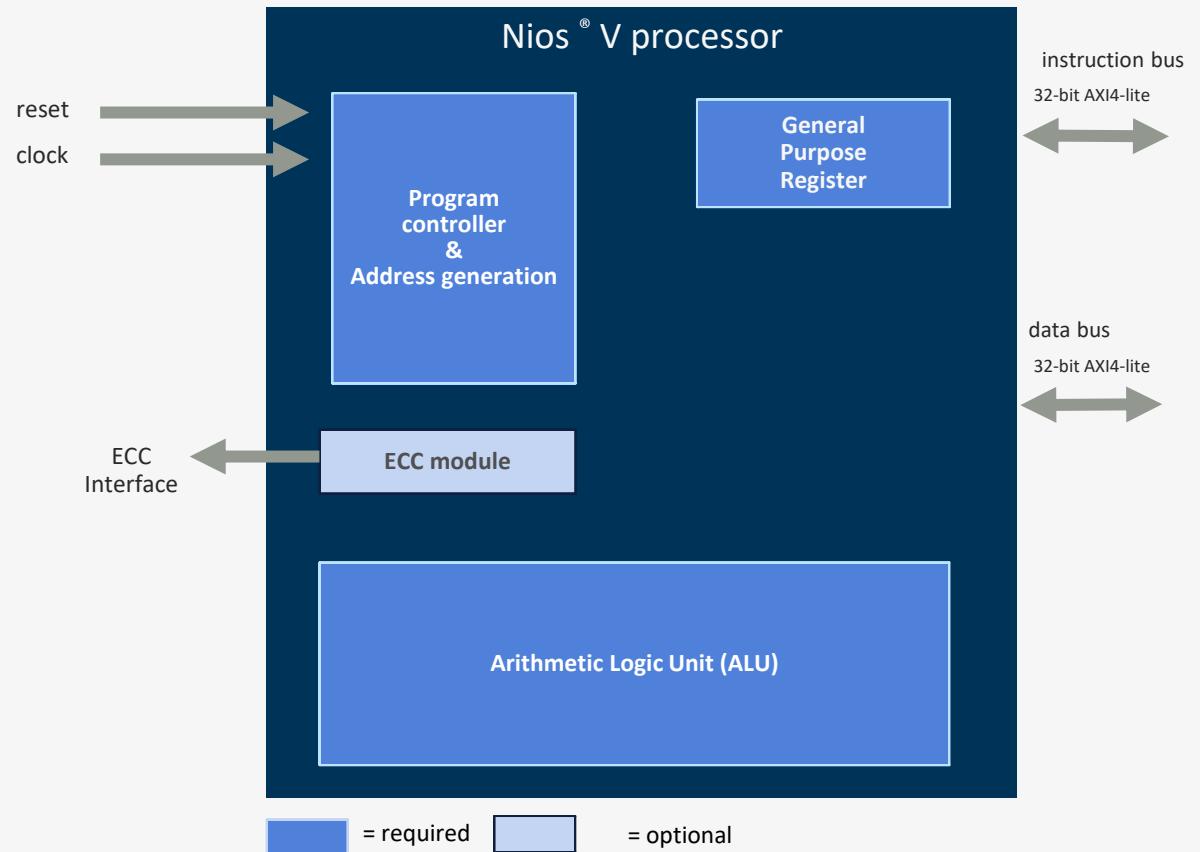
Nios® V/m Microcontroller

- Microcontroller
- RV32I Zicsr-based 32bit microcontroller
- "No pipeline" option available
 - Reduced logic capacity
 - Starting with 23.3pro/23.1std
- Supports all FPGAs



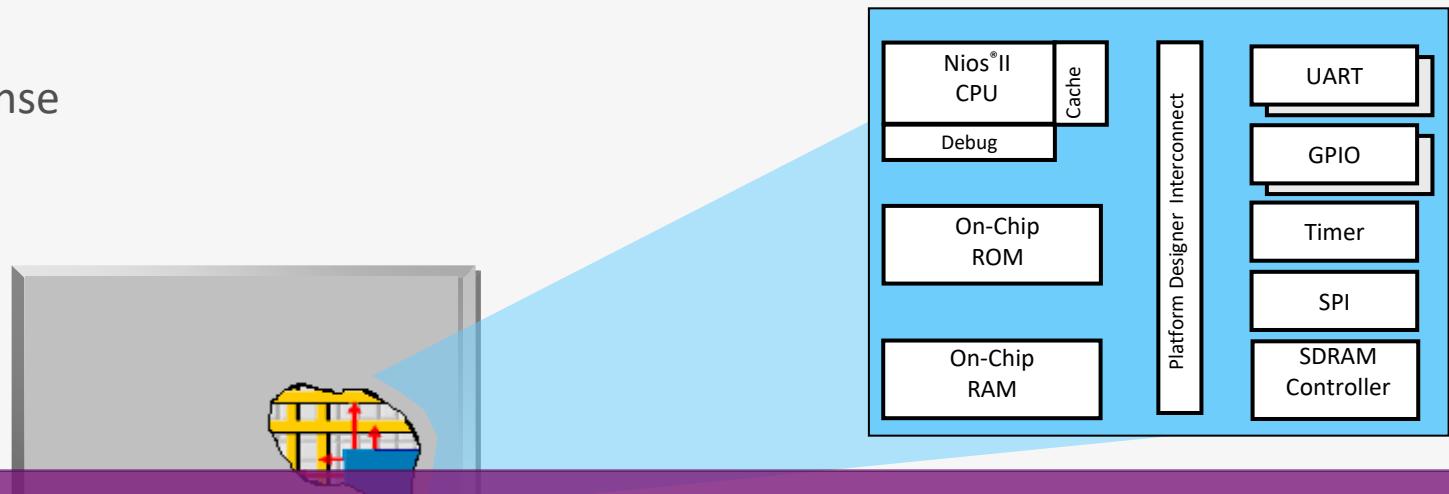
Nios® V/c Compact Microcontroller

- Compact Microcontroller
- RISC-V RV32I instruction set based
- Small footprint
 - Focus on size
 - *No debug/interrupt functionality*
 - For simple controllers
- Supports all FPGAs
 - 23.3pro/23.1 Std and later
- Supports only bare metal Intel Hardware Abstraction Layer (HAL)



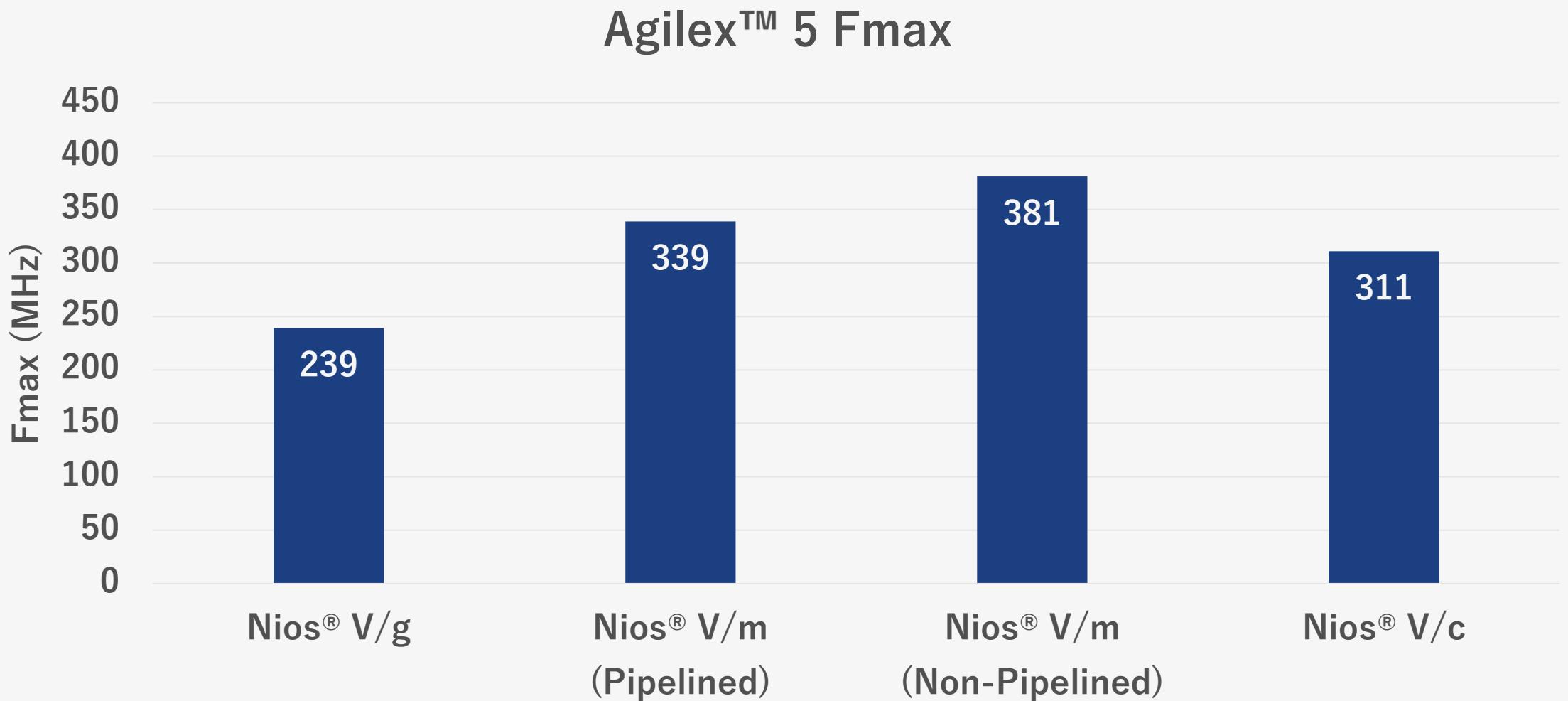
(Reference) Nios® II processor

- Soft IP 32 bit RISC processor
 - Developed independently by Altera
 - Nios® II processor and all peripherals written in HDL
 - Supports all device families
 - Synthesized with Quartus® Prime development software
 - Harvard Architecture
 - Royalty free
 - Nios® II/f requires a paid license



Nios® II available until v23.4 pro/v23.1 std
Agilex™ 5/Agilex™ 3 does not support Nios® II

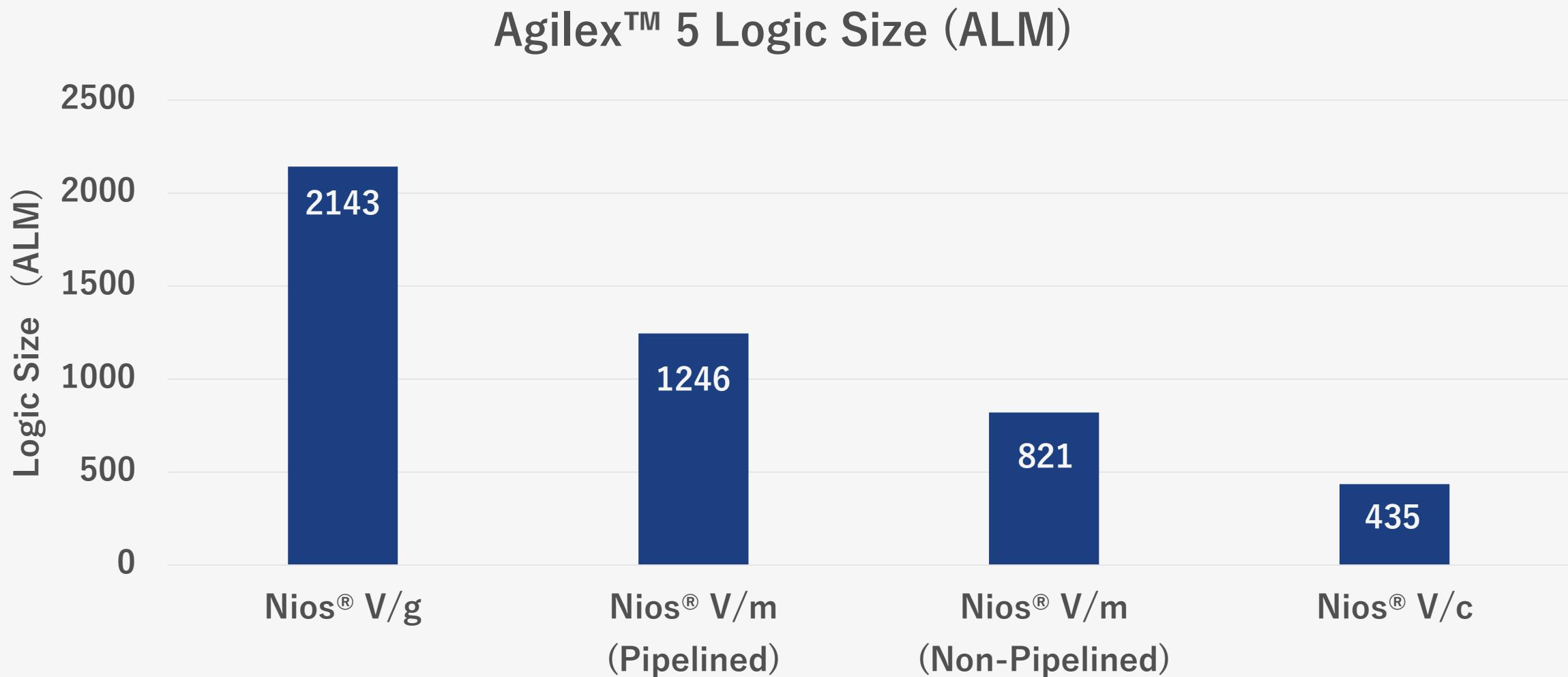
Core comparison: Fmax (Agilex™ 5)



Excerpted from [Nios® V Processor Reference Manual \(25.1\)](#)

For the latest information, refer to the [Nios® V Processor Reference Manual](#)

Core Comparison: Logic Size (Agilex™ 5)

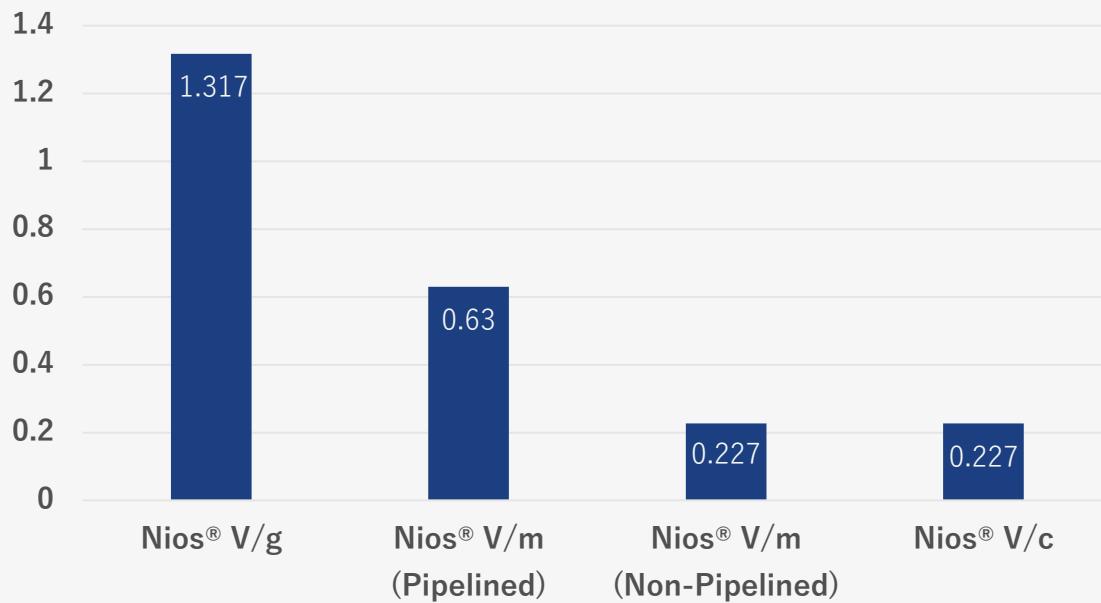


Excerpted from [Nios® V Processor Reference Manual \(25.1\)](#)

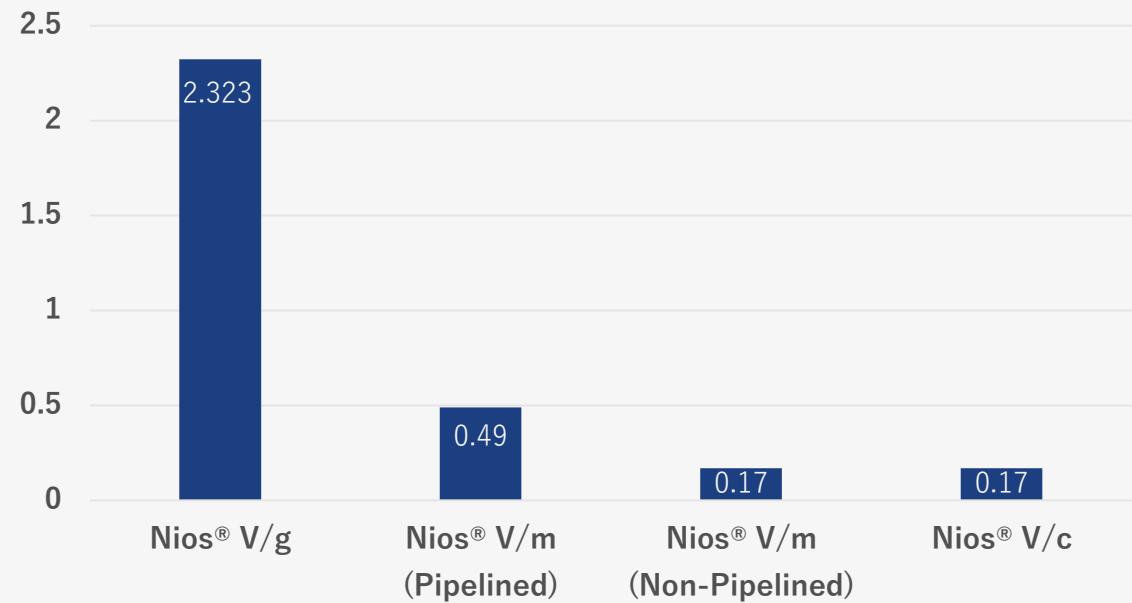
For the latest information, refer to the [Nios® V Processor Reference Manual](#)

Core Comparison: Performance (Agilex™ 5)

Agilex™ 5 DMIPS/MHz Ratio



Agilex™ 5 CoreMark/MHz Ratio



Excerpted from [Nios® V Processor Reference Manual \(25.1\)](#)

For the latest information, refer to the [Nios® V Processor Reference Manual](#)

Note: Standard Device Performance (Nios® V/g Core)

Device Family	Fmax (MHz)	Logic Size	DMIPS/MHz	CoreMark/MHz
MAX® 10	91	4199 (LE)	0.942	1.49
Cyclone® 10 LP	93	4174 (LE)	0.942	1.49
Cyclone® V	117	1886 (ALM)	0.942	1.49
Cyclone® IV E	81	4316 (LE)	0.942	1.49

Agenda

1. Agilex™ 3 FPGA & SoC FPGA Overview
2. What is Nios® V?
3. Migrating from Nios® II
4. Nios® V Hardware Development
5. Exercise 1: Hardware Exercise
6. Nios® V Software Development
7. Nios® V Boot Method
8. Introducing Debugging Tools
9. Exercise 2: Software Exercise
10. What is the FPGA AI Suite?
11. Introduction of Materials
12. Summary

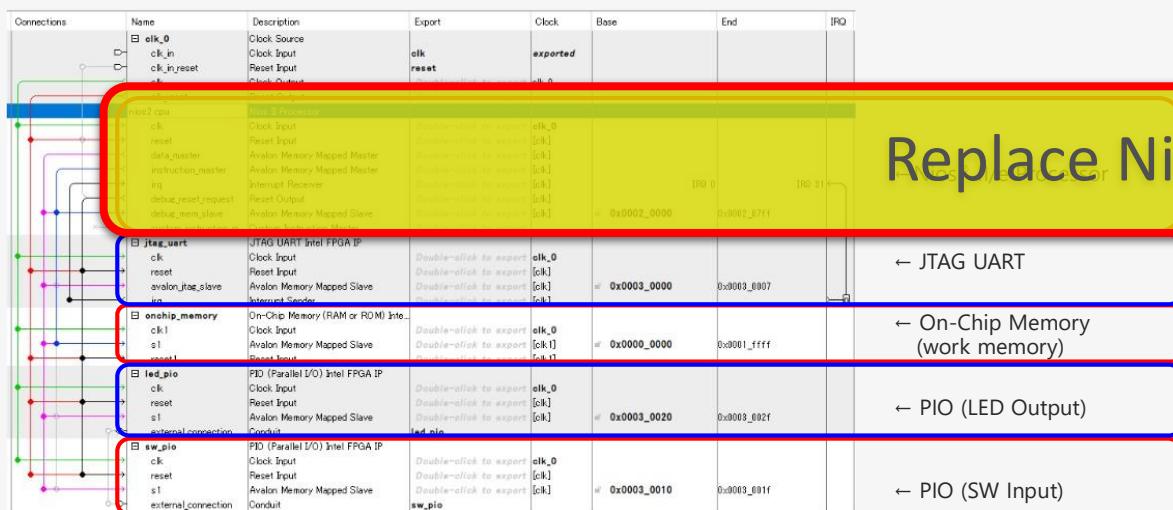
Supported Processors

Nios® II Processors	Nios® V Processors	Quartus® Prime Release Version	
		Pro	Standard
Nios® II/e Processors (No JTAG debugging, No interrupts and exceptions)	Nios® V/c Processor	23.3	23.1
Nios® II/e Processor	Nios® V/m Processor (no pipeline)	23.3	23.1
Nios® II/e Processor			
Nios® II/f Processor (no multipliers, no cache, no custom instructions)	Nios® V/m processor	21.3	22.1
Nios® II/f processor (with multiplier/divider, with cache, with custom instructions)	Nios® V/g processor	23.1	23.1

Migration: Hardware

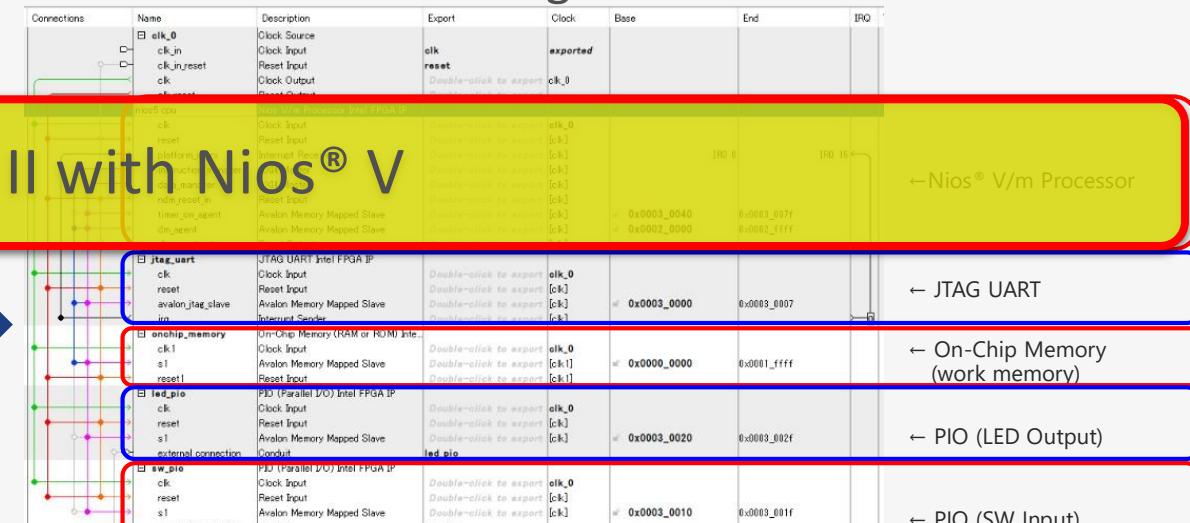
- Basically, all you need to do is change Nios® II to Nios® V
 - See [AN 978: Nios® V Processor Migration Guide](#) for details on the changes
- Interface changed to AXI-4, but you can connect to the old Avalon-MM Slave
 - Avalon-MM/AXI conversion is supported by Platform Designer
 - The previous Slave design remains usable without modification
 - Nios®/m and Nios® V/c can also select Avalon-MM

Nios® II design



Replace Nios® II with Nios® V

Nios® V design



Migration: Software

- Relatively easy migration if using HAL API
- If using OS, contact each vendor
- If using assembler, customer needs to modify
- Different way to create software project
 - Will be introduced later in presentation
 - See the following articles for more details
 - [Nios®V Project Development Procedure Using Ashling® RiscFree® IDE](#)

AN 978: Nios® V Processor Migration Guide

AN 978: Nios® V Processor Migration Guide

[Download ↓](#)[Bookmark ▾](#)[View More ▾](#) Search this document

- 1. Overview
- 2. Architecture Comparison
- 3. Processor Core Migration Mapping
- 4. Design Flow Comparison
- 5. Migration by Use Case
- 6. References
- 7. Document Revision History for AN 978: Nios® V Processor Migration Guide

4. Design Flow Comparison

Table 4. Migration Consideration by Design Flow

Design Stage	Nios® V Processor	Nios® II Processor	Migration Consideration
Intel® Quartus® Prime Project Creation	Create a new project using the New Project Wizard .	Create a new project using the New Project Wizard .	None
Define and Generate System in the Platform Designer	1. Instantiate the Nios® V processor core. 2. Create a Nios® V processor system with basic peripherals.	1. Instantiate the Nios® II processor core. 2. Create a Nios® II processor system with basic peripherals.	1. Nios® V processor has a similar interface as the Nios® II processor. You can replace the interface in the Platform Designer. 2. Refer to the Table <i>Core Migration</i> for the processor core mapping guidelines. 3. Refer Table <i>Primary Interface</i> for the signals connection mapping guidelines.

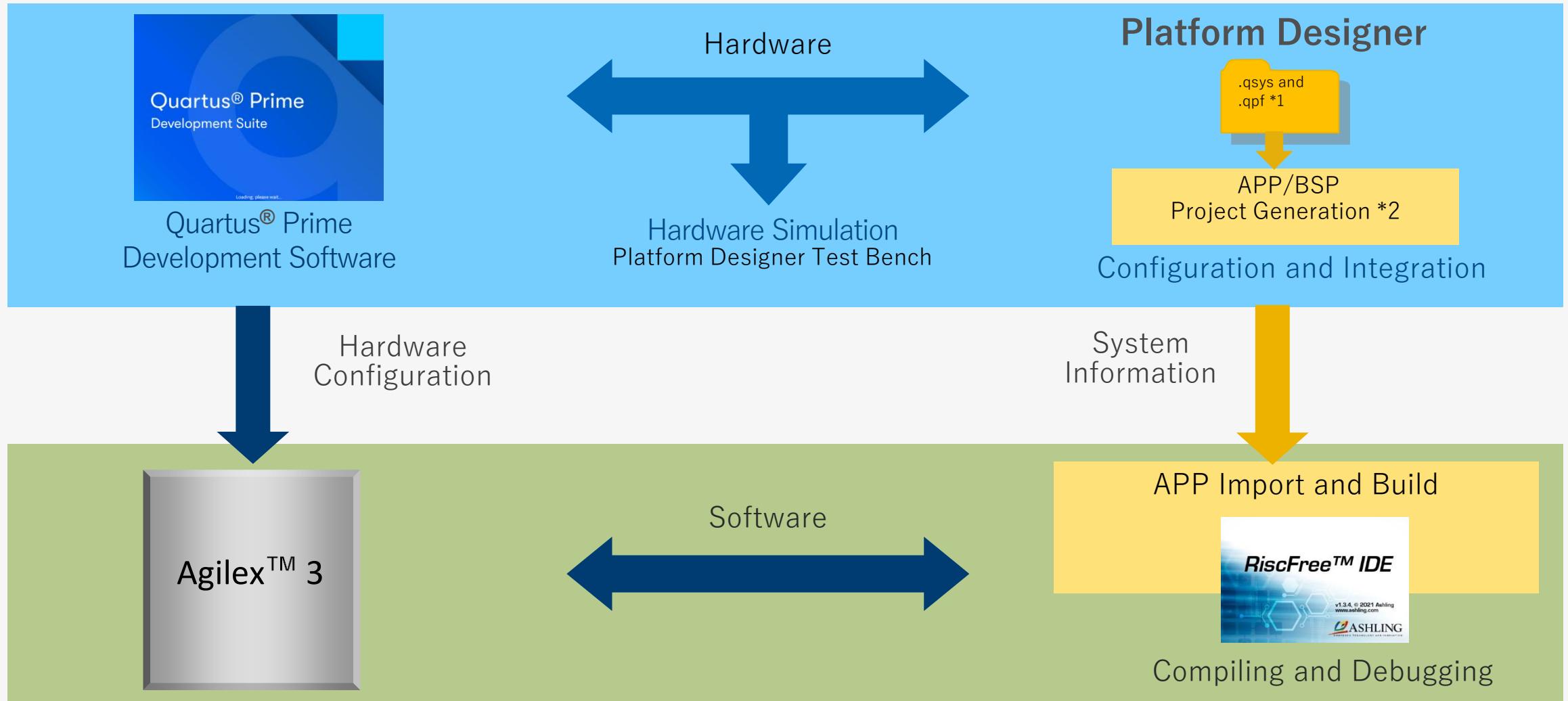
See [AN978](#) for details on the differences between Nios® II and Nios® V.

<https://www.intel.com/content/www/us/en/docs/programmable/773196/current/design-flow-comparison.html>

Agenda

1. Agilex™ 3 FPGA & SoC FPGA Overview
2. What is Nios® V?
3. Migrating from Nios® II
4. Nios® V Hardware Development
5. Exercise 1: Hardware Exercise
6. Nios® V Software Development
7. Nios® V Boot Method
8. Introducing Debugging Tools
9. Exercise 2: Software Exercise
10. What is the FPGA AI Suite?
11. Introduction of Materials
12. Summary

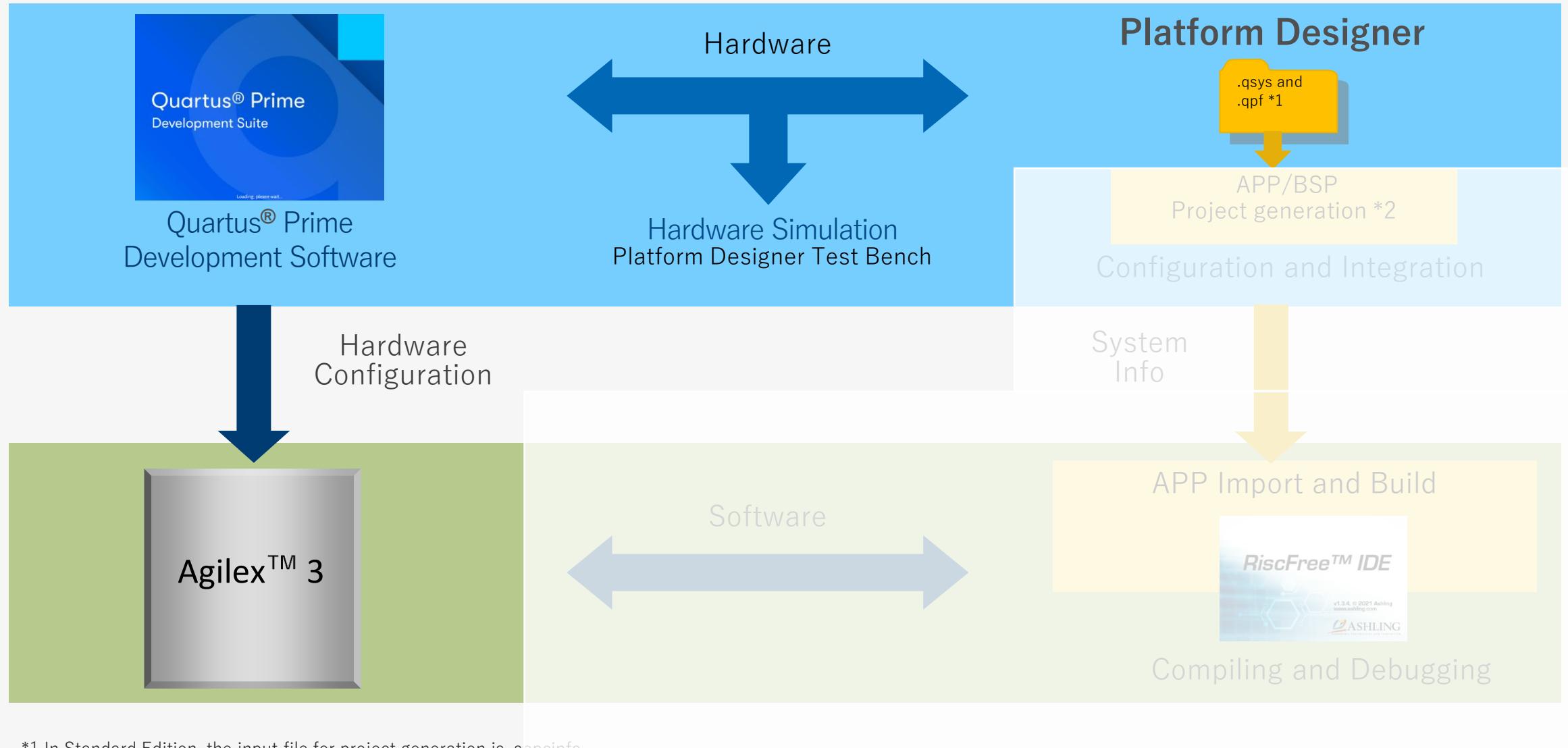
Nios® V Processor Development Flow



*1 For Standard Edition, the input file for project generation is .sopcinfo.

*2 BSP generation is supported by both the command line tool and the BSP editor on Platform Designer.

Nios® V Processor: Hardware Development Flow



*1 In Standard Edition, the input file for project generation is .sopcinfo.

*2 BSP generation is supported by both the command line tool and the BSP editor on Platform Designer.

What is Quartus® Prime?

- Software (tools) for Altera® FPGA development
- This one^(*) covers the work required for development
 - Logic circuit design and programming (writing to the device)

Functional simulation only Implemented with Questa -Altera® FPGA Edition



- Built-in debugging function for the actual device
 - Signal Tap logic analyzer, etc.
- < Reference > Let's try Agilex™ 3 FPGA Debug "Signal Tap Logic Analyzer"
- <https://malt.zendesk.com/hc/articles/49591414544281>

Quartus® Prime Editions Overview

Agilex™ 3 uses Pro Edition

	Lite Edition	Standard Edition	Pro Edition
	Download	Download	Download
Support Device	Agilex™ 7		✓
	Agilex™ 5, Agilex™ 3		✓※2
	Stratix® 10		✓
	Arria® 10	✓※1	✓
	Cyclone® 10 GX		✓※2
	Cyclone® 10 LP	✓	✓
	Stratix® V, Stratix® IV		✓
	Arria® II	✓ (partial)	✓
	Cyclone® V, Cyclone® IV	✓	✓
	MAX® series	✓	✓
Function		See below	
Cost	License	Free No License Required Paid	License Required (* 1: For new development of Arria® 10, Pro Edition is recommended) Chargeable License required (* 2: Agilex™ 5, Agilex™ 3, Cyclone® 10 GX available free of charge)

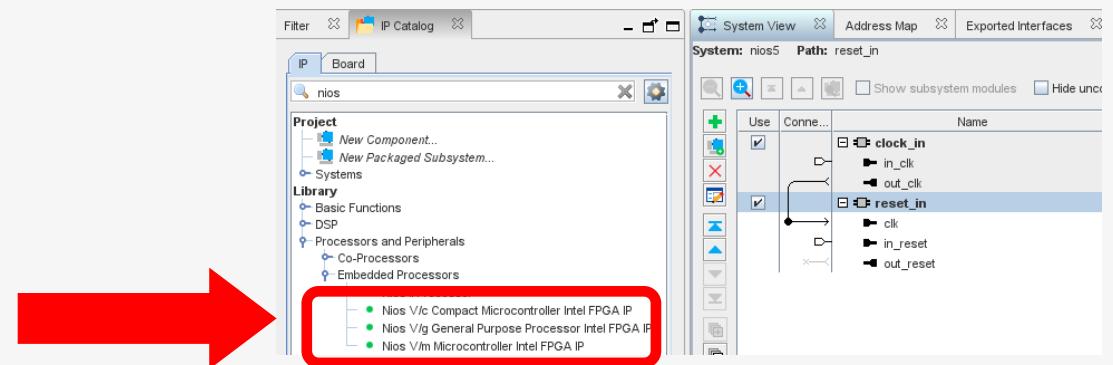
- Different editions support different device families and features

 - See Quartus® Prime Design Software - Compare PRO, STANDARD and LITE Editions

<https://www.intel.com/content/dam/www/central-libraries/us/en/documents/quartus-prime-compare-editions-guide.pdf>

Nios® V Processor Hardware Development: Platform Designer

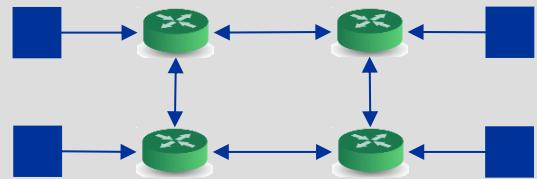
- Simple and standard design flow similar to Nios® II
- Quickly build systems with Platform Designer
- Select from IP Catalog
 - IP Catalog
 - > Embedded Processor
 - > Nios V/c Compact Microcontroller Intel FPGA IP
 - Nios V/g General Purpose Processor Intel FPGA IP
 - Nios V/m Microcontroller Intel FPGA IP



Platform Designer Overview

- Auto-generation Tool
 - High-performance interconnect
 - Layered design
 - Industry-standard interface support
 - IP management capabilities
 - Real-time system debugging capabilities

High-Performance Interconnect

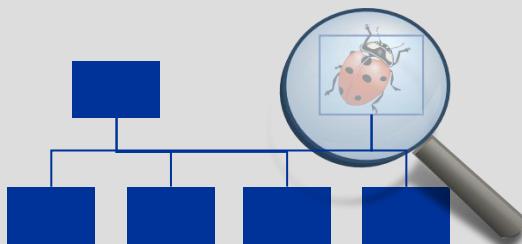


Based on Network-on-chip Architecture

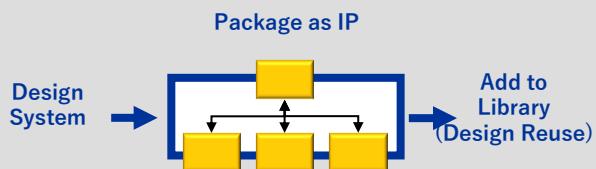
Industry-standard Interfaces

AMBA	AXI3, AXI4, AXI4-Lite, AXI4-Stream ACE-Lite, ACE5-Lite APB
Avalon®	Avalon-MM Avalon-ST

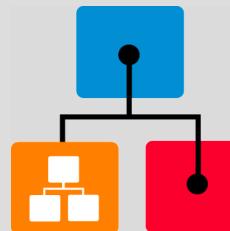
Real-time System Debug



IP Management



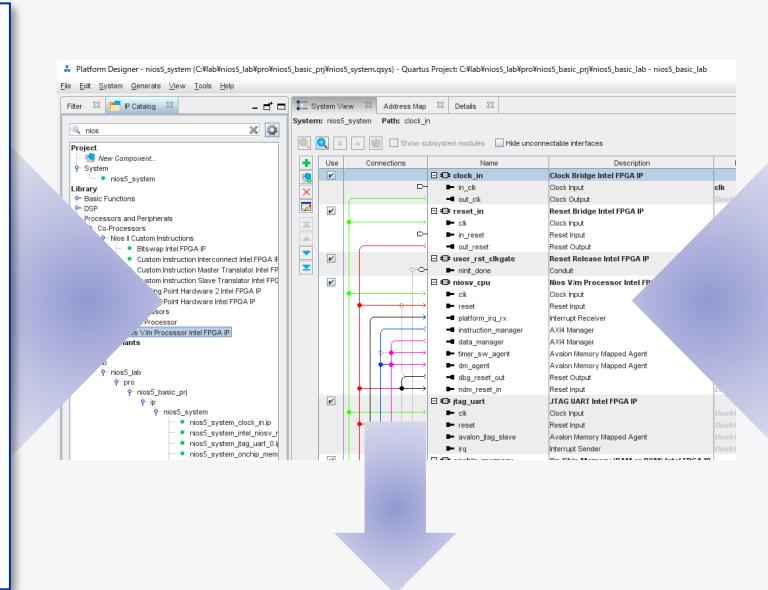
Hierarchy



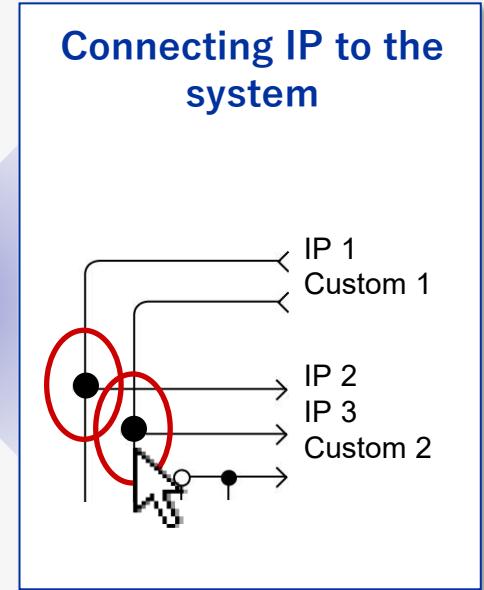
Overview of Platform Designer

IP library

- Interface protocols
- Memory
- DSP
- Embedded
- Bridges
- PLL
- Custom systems



Connecting IP to the system



Accelerate development



Simplified implementation

Automate error-prone implementation tasks

Customize - Choose only what you need

The screenshot shows the Quartus Project Manager interface. On the left, the 'IP Catalog' tab is selected, displaying a search bar and a tree view of available IP components categorized by project, library, and existing variants. A red box highlights this area. On the right, the 'System View' tab is selected, showing a system diagram for 'nios5_system'. The diagram includes nodes like 'clock_in', 'reset_in', 'user_rst_in', 'niosv_cpu', and 'jtag_uart', with their connections and descriptions. A blue box highlights the system view area.

Altera® FPGA, Partner & User Core

- Processor
- Memory Interface
- Peripherals
- Bridges
- Importing User Logic

Easily connect and delete peripherals! Easy to redesign!

Nios® V setup screen

Nios® V/g

Nios V/g General Purpose Processor Intel FPGA IP - intel_niosv_g_0

MegaCore® intel_niosv_g

Example Designs For available Nios V/g Example Designs, please go to: Quartus > File Menu > Open Example Project.

CPU Architecture

- Enable Branch Prediction
- Enable Cache
- Enable Debug
- Enable Reset from Debug Module

Lockstep

- Enable Lockstep
- Blind Window Period
- Default Timeout Period
- Enable Interlocked Reset Interface

Use Reset Request

- Add Reset Request Interface

Trap, Exceptions, and Interrupts

- Reset Agent: Absolute
- Reset Vector Offset: 0x00000000
- Enable Core Level Interrupt Controller

Interrupt Mode

- Direct
- Disabled

Shadow Register Files

- Caches
- Data Cache Size: 4 Kilobytes
- Instruction Cache Size: 4 Kilobytes

Peripheral Regions

- Peripheral Region A | Peripheral Region B
- Peripheral Region A TCM
- Peripheral Region A Base Address: 0x00000000
- Peripheral Region TCM Initialization File

Tightly Coupled Memories

- Instruction TCM | Instruction TCM2 | Data TCM | Data TCM2
- Instruction TCM2 Size: 0
- Instruction TCM2 Base Address: 0x00000000
- Instruction TCM2 Initialization File

ECC

- Enable Error Detection & ECC Status Reporting
- Enable Single Bit Correction

Custom Instructions

Hardware Instructions Table

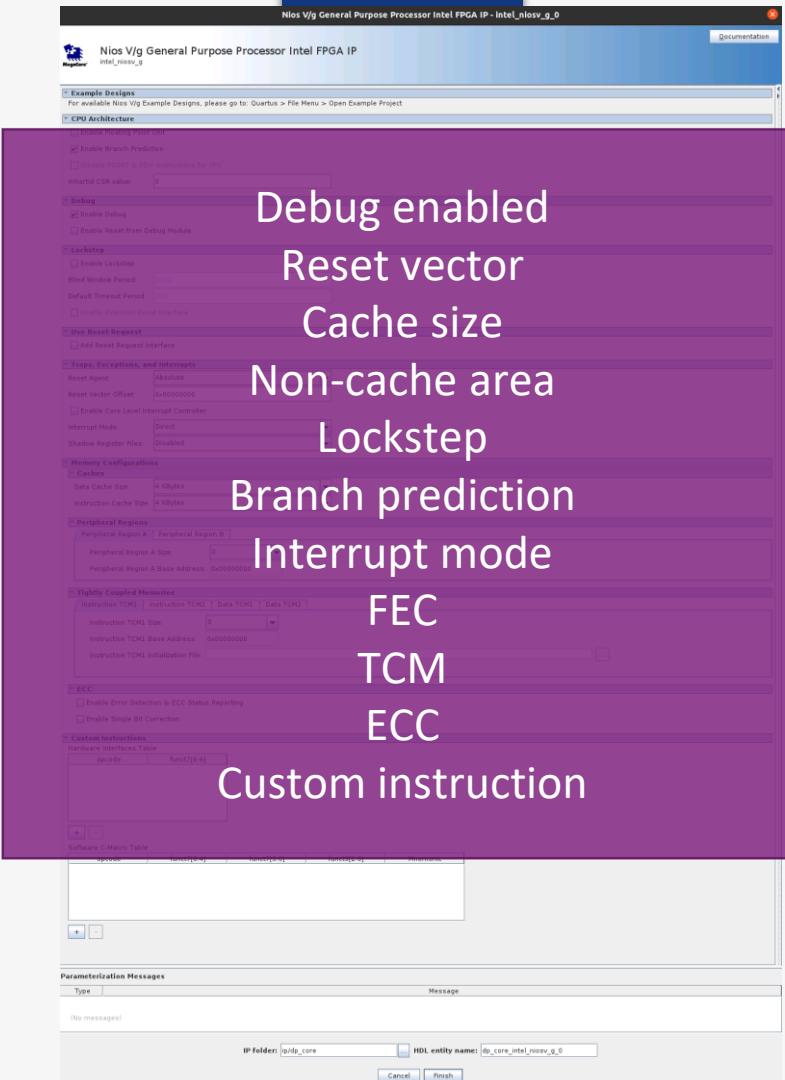
Software C-Macro Table

Parameterization Messages

Type	Message
(No messages)	

IP folder: ip/dp_core HDL entity name: ip_c_core_intel_niosv_g_0

Cancel Finish



Nios® V /m

Nios V/m Microcontroller Intel FPGA IP - intel_niosv_m

MegaCore® intel_niosv_m

Example Designs For available Nios V/m Example Designs, please go to: Quartus > File Menu > Open Example Project.

Debug

- Enable Debug
- Enable Reset from Debug Module

Use Reset Request

- Add Reset Request Interface

Traps, Exceptions, and Interrupts

- Reset Agent: Absolute
- Reset Vector Offset: 0x00000000
- Interrupt Mode: Direct

CPU Architecture

When pipelining is enabled, you benefit from higher performance at the cost of higher area and lower frequency. When pipelining is disabled, you benefit from lower area and higher frequency, but at the cost of lower IPC.

- Enable Pipelining in CPU
- Enable Avalon Interface

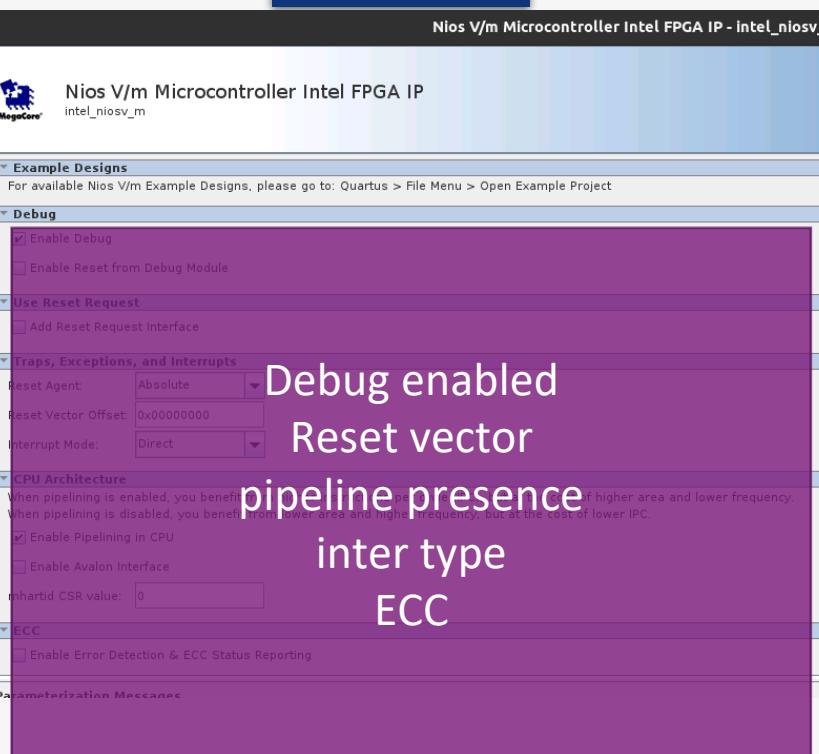
mhartid CSR value: 0

ECC

- Enable Error Detection & ECC Status Reporting

Parameterization Messages

Type	Message
(No messages)	



Nios® V /c

Nios V/c Compact Microcontroller Intel FPGA IP - intel_niosv_c

MegaCore® intel_niosv_c

Example Designs For available Nios V Example Designs, please go to: Quartus > File Menu > Open Example Project https://www.intel.com/content/www/us/en/support/programmable/support-resources/design

CPU Architecture

- Enable Avalon Interface

mhartid CSR value: 0x00000000

Use Reset Request

- Add Reset Request Interface

Traps, Exceptions, and Interrupts

- Reset Agent: Absolute
- Reset Vector Offset: 0x00000000

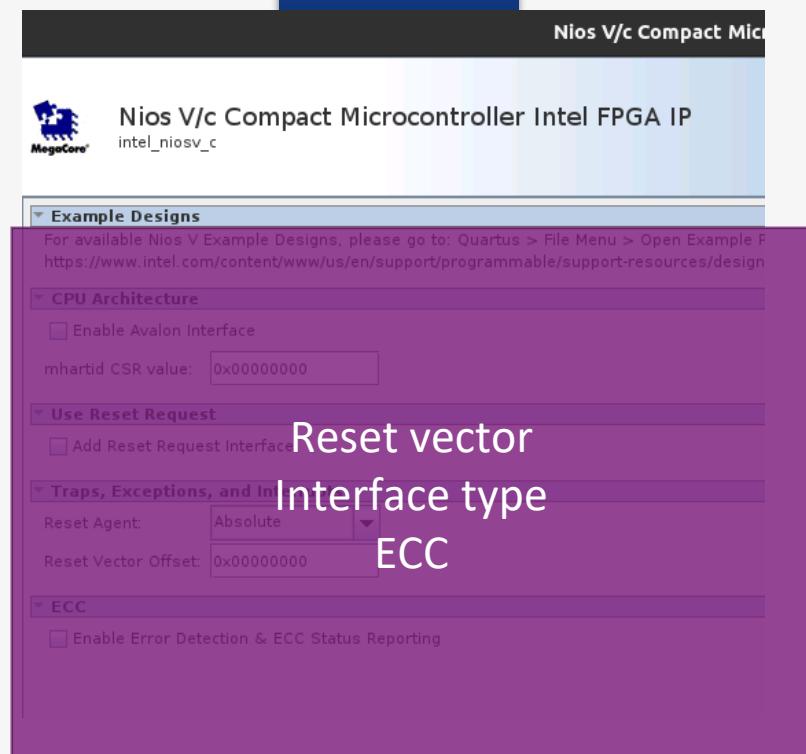
Interface type

- ECC

Enable Error Detection & ECC Status Reporting

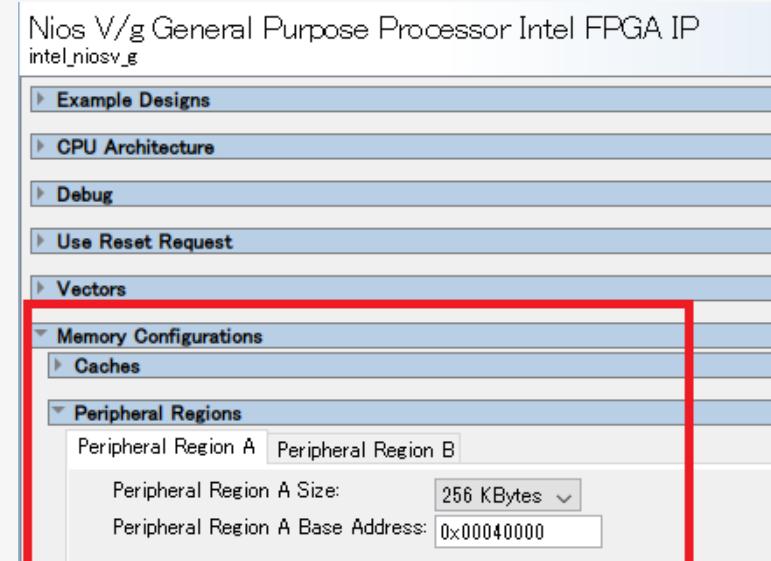
Parameterization Messages

Type	Message
(No messages)	

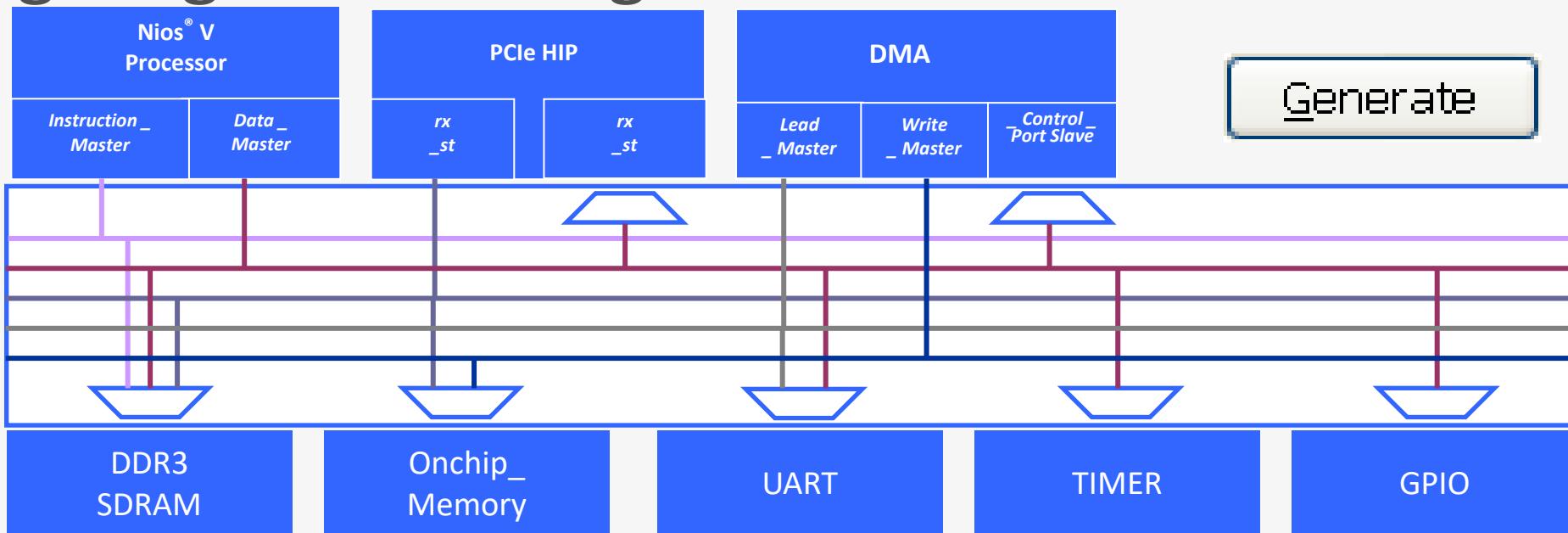


Accessing non-cache regions (Nios® V/g)

- For non-cache access, the target region must be specified.
 - Set as a peripheral IO region (Peripheral Region) in Platform Designer.
- Setting method
 - Set the region for non-cache access in Memory Configurations -> Peripheral Regions.
 - Two regions can be set.
 - Example: Set 256KB from 0x0004_0000 to 0x0007_FFFF as non-cache region.
- Also refer to the following content
 - [How to do non-cache access with Nios® V](#)
 - [AN 978: Nios® V Processor Migration Guidelines](#)



Building using Platform Designer

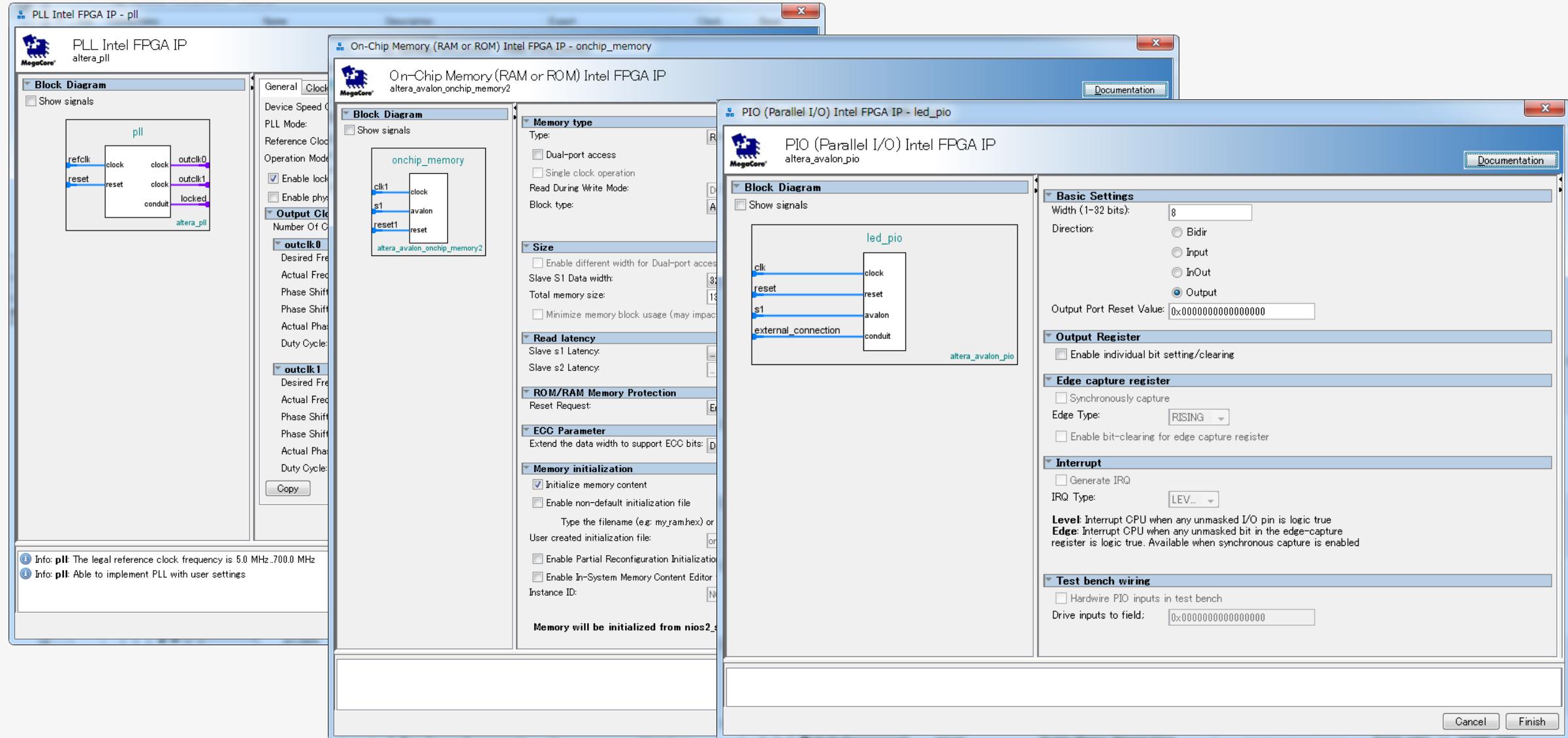


Use	Connections	Name	Description	Export	Clock	Base	End	IRQ
<input checked="" type="checkbox"/>		Nios_II data_master instruction_master	Nios II Processor Avalon Memory Mapped Master Avalon Memory Mapped Master	nios_ii_data_master nios_ii_instruction_master	[clk] clk_0 [clk1]		IRQ 0	IRQ 31

Platform Designer creates bus connections inside the FPGA
all automatically.

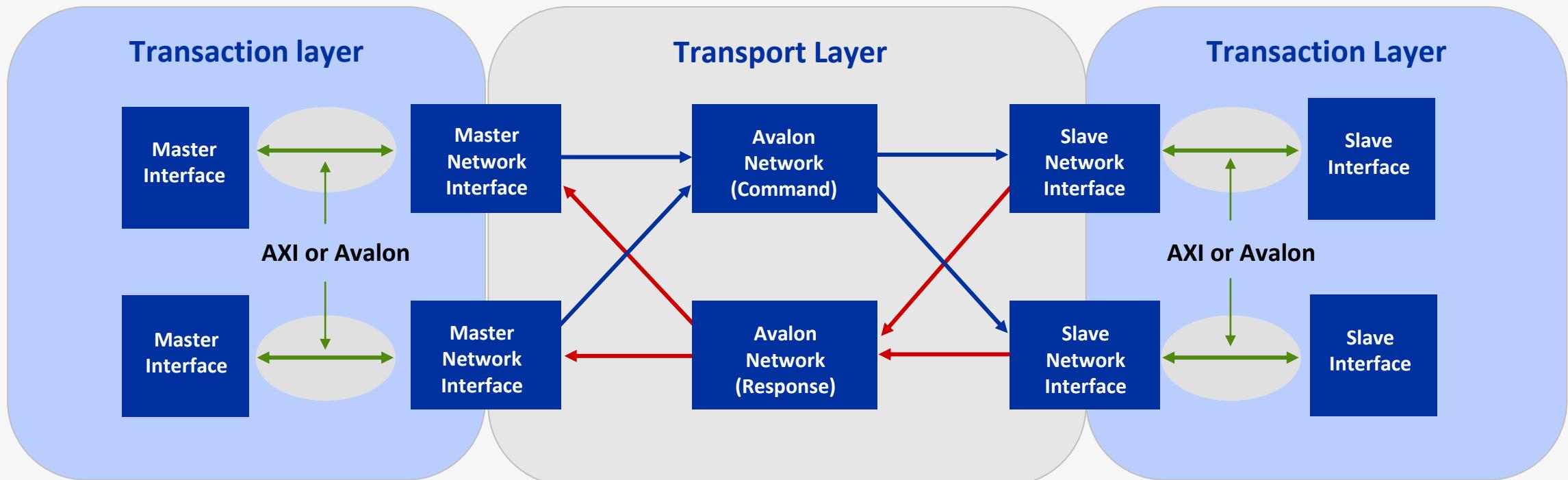


Customizing - Specifying the Specs You Want



Conversion between AXI4 and Avalon-MM Slave

- Conversion of interfaces using Platform Designer
- Network-on-Chip (NoC) architecture
 - Transaction layer and transport layer
 - Encapsulate each command in a packet and forward to slave
 - Encapsulate each response in a packet and return to master

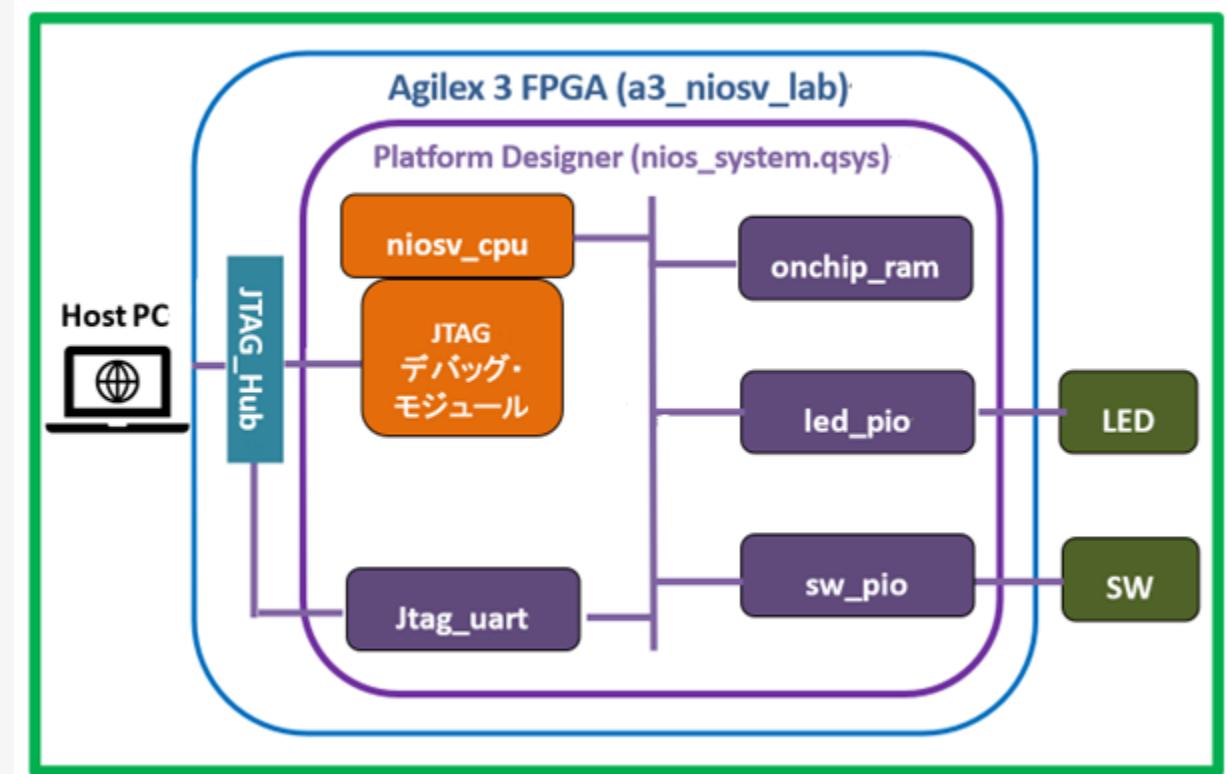
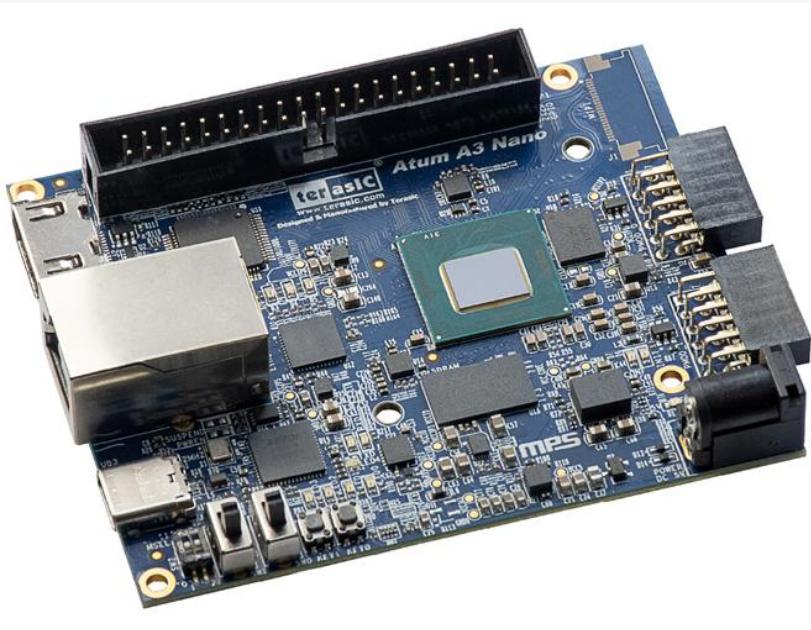


Agenda

1. Agilex™ 3 FPGA & SoC FPGA Overview
2. What is Nios® V?
3. Migrating from Nios® II
4. Nios® V Hardware Development
5. Exercise 1: Hardware Exercise
6. Nios® V Software Development
7. Nios® V Boot Method
8. Introducing Debugging Tools
9. Exercise 2: Software Exercise
10. What is the FPGA AI Suite?
11. Introduction of Materials
12. Summary

Exercise 1: Hardware Exercise

- Implement Nios® V on Agilex™ 3 and "Hello world" and "LED Blinking"
 - In Exercise 1, implement Nios® V hardware on an already created FPGA project
- Equipment Used
 - [Terasic - All FPGA Boards - Agilex 3 - Atum A3 Nano](#)



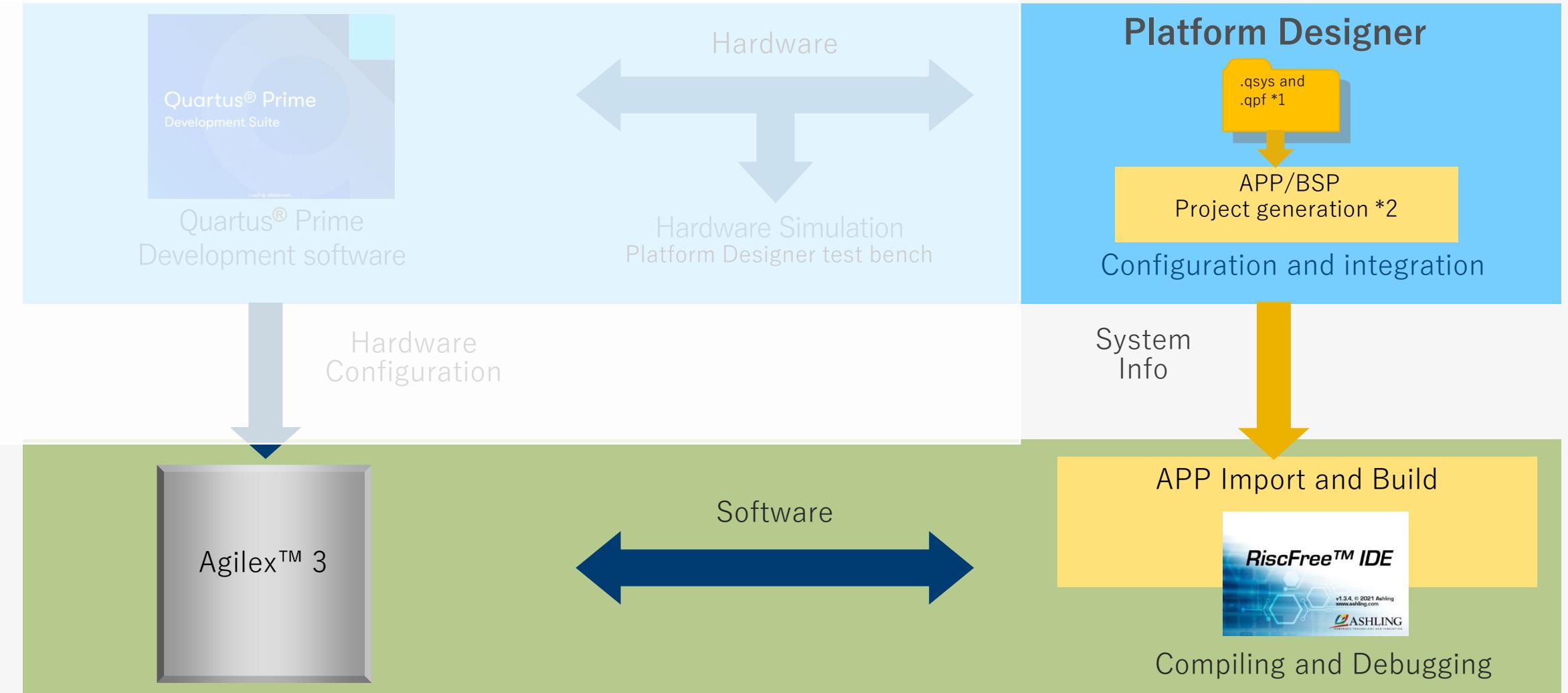
Hardware Design

Use	Connections	Name	Description	Export	Clock	Base	End	IRQ
✓		clock_in	Clock Bridge Intel FPGA IP Clock Input Clock Output	clk	exported <i>Double-click to export</i>	clock_in_out...		
✓		reset_in	Reset Bridge Intel FPGA IP Clock Input Reset Input Reset Output	reset	<i>Double-click to export</i>	clock_in_o...		
✓		pll	IOPLL FPGA IP Clock Input Conduit Reset Input	Double-click to export pll_locked	clock_in_o...			
✓		cpu	Nios V/g General Purpose Processor Intel F... Clock Input Reset Input Interrupt Receiver AXI4 Manager AXI4 Manager Avalon Memory Mapped Agent Avalon Memory Mapped Agent Reset Output Reset Input	Double-click to export Double-click to export	pll_outclk0 [clk]	IRQ 0 0x0083_0000 0x0082_0000	IRQ 15 0x0083_003f 0x0082_ffff	
✓		onchip_memory	On-Chip Memory II (RAM or ROM) IP Clock Input Avalon Memory Mapped Agent Reset Input	Double-click to export Double-click to export Double-click to export	pll_outclk0 [clk1]	0x0000_0000 0x0001_ffff		← On-Chip Memory (work memory)
✓		jtag_uart	JTAG UART IP Clock Input Reset Input Avalon Memory Mapped Agent	Double-click to export Double-click to export Double-click to export	pll_outclk0 [clk]	0x0083_0070 0x0083_0077		← JTAG UART
✓		led_pio	PIO (Parallel I/O) IP Clock Input Reset Input Avalon Memory Mapped Agent Conduit	Double-click to export Double-click to export Double-click to export Double-click to external connect...	pll_outclk0 [clk]	0x0083_0060 0x0083_006f		← PIO (LED Output)
✓		sw_pio	PIO (Parallel I/O) IP Clock Input Reset Input Avalon Memory Mapped Agent Conduit	Double-click to export Double-click to export Double-click to export sw_pio_external_connect...	pll_outclk0 [clk]	0x0083_0050 0x0083_005f		← PIO (SW Input)

Agenda

1. Agilex™ 3 FPGA & SoC FPGA Overview
2. What is Nios® V?
3. Migrating from Nios® II
4. Nios® V Hardware Development
5. Exercise 1: Hardware Exercise
6. Nios® V Software Development
7. Nios® V Boot Method
8. Introducing Debugging Tools
9. Exercise 2: Software Exercise
10. What is the FPGA AI Suite?
11. Introduction of Materials
12. Summary

Nios® V Processor: Software Development Flow

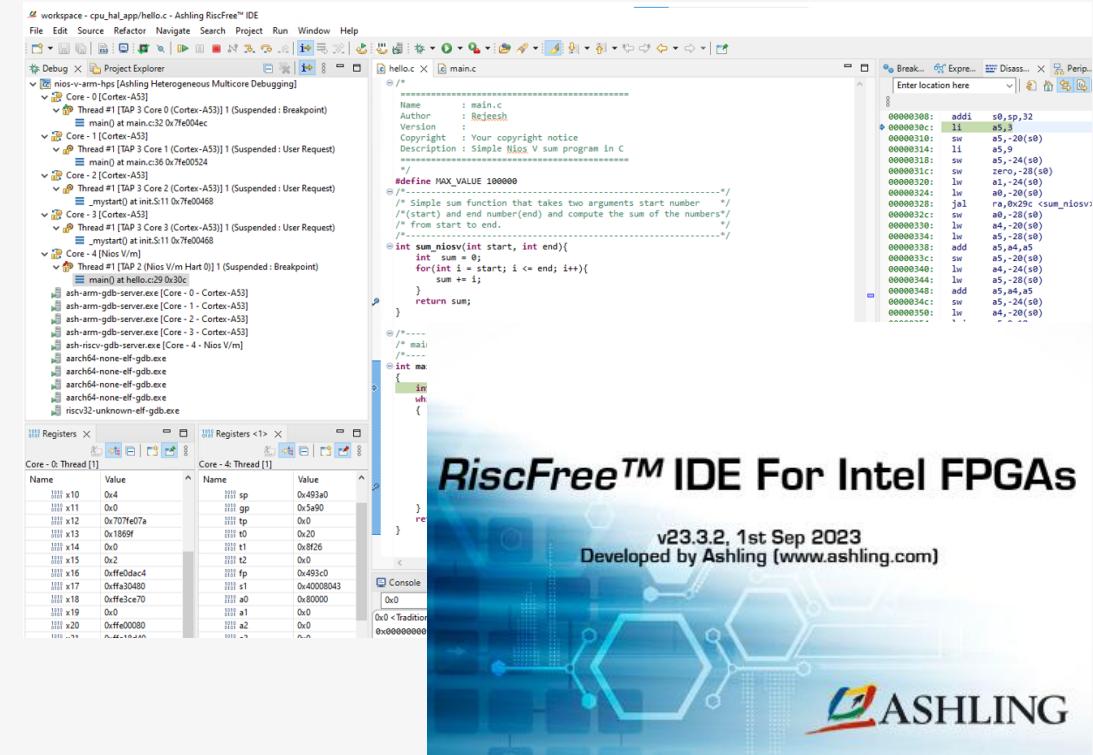


*1 In Standard Edition, the input file for project generation is .sopcinfo.

*2 BSP generation is supported by both the command line tool and the BSP editor on Platform Designer.

Ashling RiscFree IDE for Altera® FPGAs

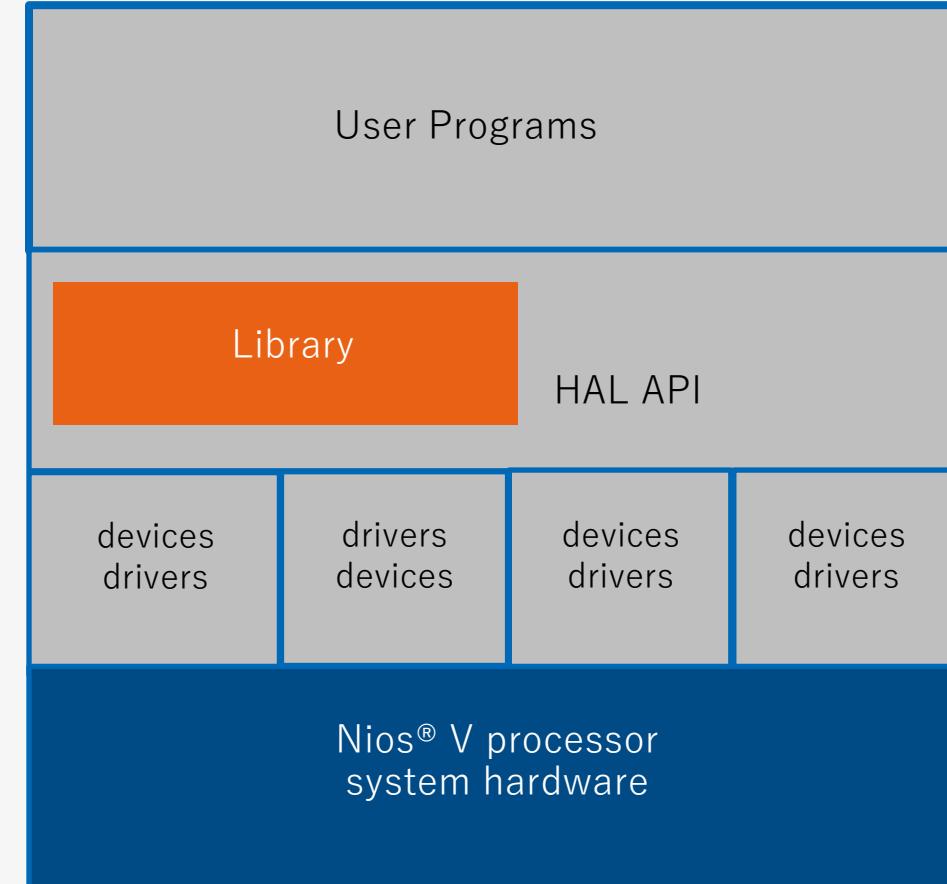
- Available from the Quartus® Prime download site
 - Pro Edition version 22.2 and later
 - Standard Edition version 22.1 and later
 - Also available as a separate free application for embedded software developers
 - Integration of various toolchains
 - Ashling RiscFree IDE for Altera® FPGA
 - Compiler
 - Debugger
 - Tracing
 - Software development and debugging support for Nios® V and Arm processors



Development out of the box with Ashling's RiscFree IDE for Altera® FPGAs

Software Development for the Nios® V Processor: Software Stack

- Altera Hardware Abstraction Layer (HAL)
 - Interface with the Nios® V Processor
- Third-Party Real-Time Operating Systems (RTOS)
 - Micrium MicroC/OS-II
 - FreeRTOS
 - Zephyr
 - More supported OSes will be added in the future



Software Development Flow



✓ Using the BSP Editor launched from Platform Designer (*)

✓ Run niosv-app in the Nios® V Command Shell

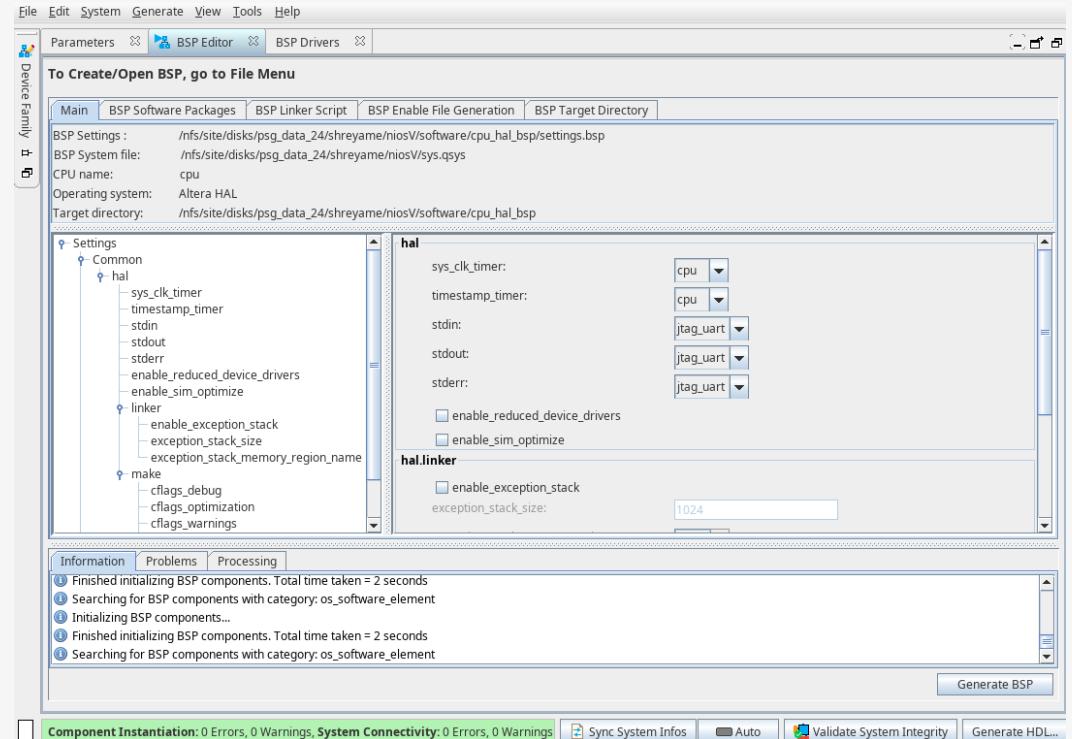
✓ Ashling RiscFree IDE for Intel® FPGA

✓ Ashling RiscFree IDE for Intel® FPGA

- Two Software Projects
 - Board Support Package (BSP) Project
 - Application (APP) Project

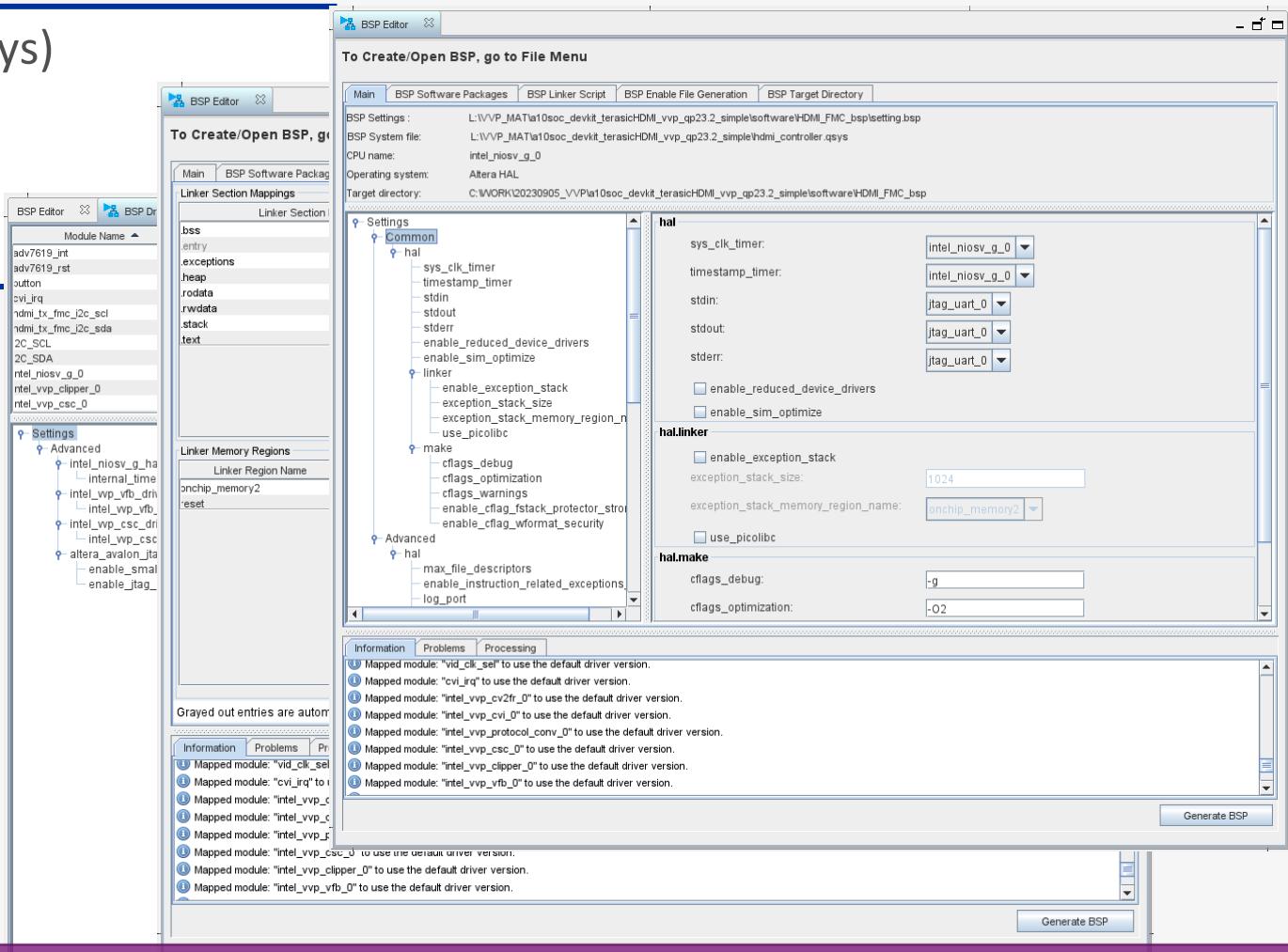
Board Support Package (BSP)

- Altera provides a BSP generation tool
 - Same as Nios® II
- Two generation methods
 - GUI of the BSP editor
 - Command line interface
- Allows generation of BSPs for the latest design configurations



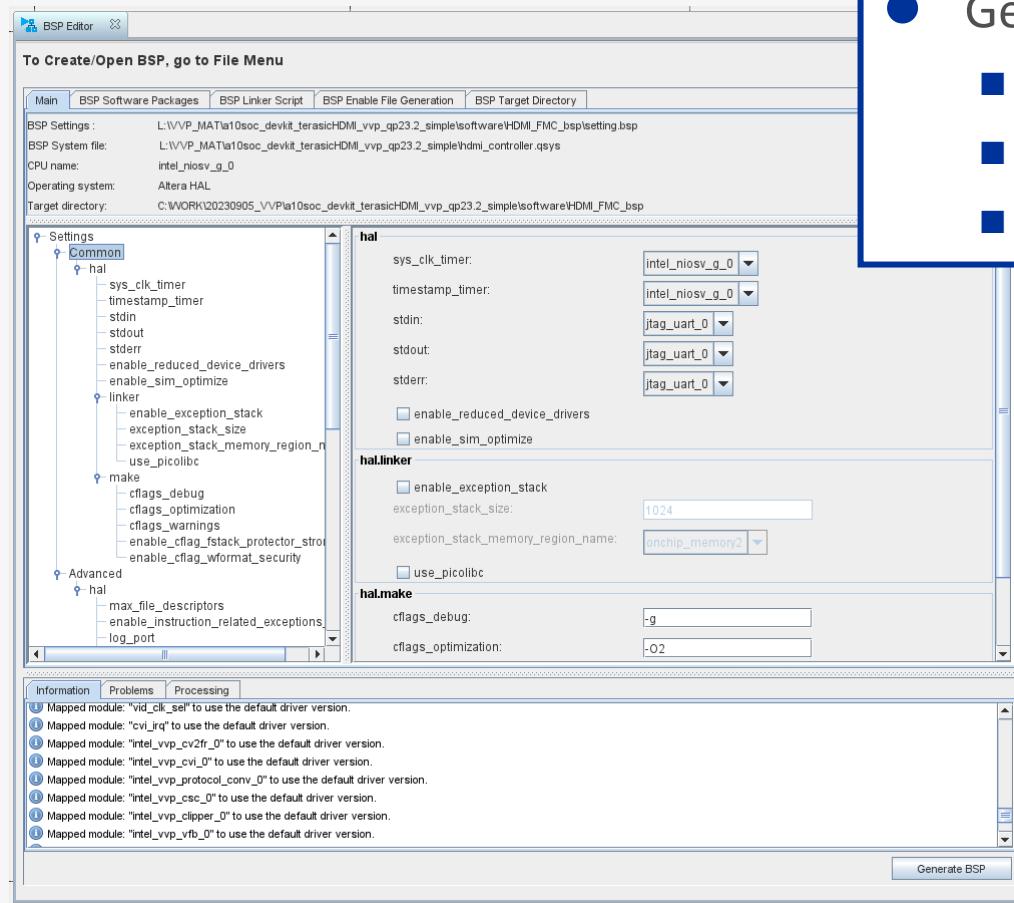
Board Support Package (BSP)

- Import System Information File (.qsys)
 - IP (Driver) Information
 - Linker Settings
 - Other Various Options



Import .qsys file (*) to import HW design information required for SW development

Creating a BSP Project



- Generate generates a BSP project that includes:
 - Various Drivers
 - A system.h file that contains HW information
 - A Linker script file that contains SW placement information



You can create a BSP project simply using the BSP Editor!

Creating an APP project

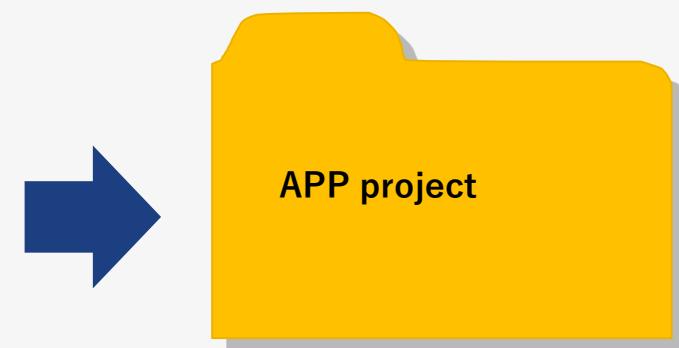
- The following command creates an APP project and generates CMakeLists.txt.

① Creating an APP project	
Commands	niosv-app --bsp-dir= < directory of bsp project > --app-dir= < directory of app project > --srcs= < directory of source code > --elf-name= < name of ELF file to be created >
Execution example	niosv-app --bsp-dir=software/hal_bsp --app-dir=software/app --srcs=software/app/led_output.c --elf-name=led_output.elf

```
[niosv-shell] C:\$work\%nios5_basic_lab> niosv-app --bsp-dir=software/hal_bsp --app-dir=software/app --srcs=software/app/led_output.c --elf-name=led_output.elf  
2022.12.15.15:11:08 Info: Elf_name is set to "led_output.elf".  
2022.12.15.15:11:08 Info: "software\app\%CMakeLists.txt" was generated.
```

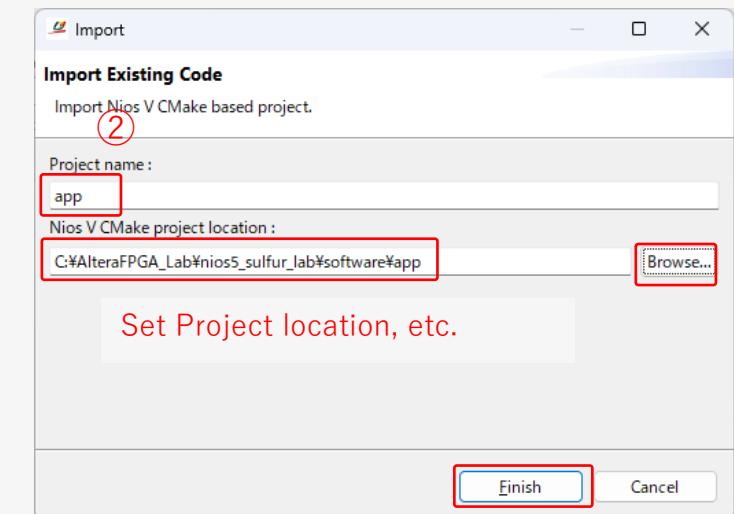
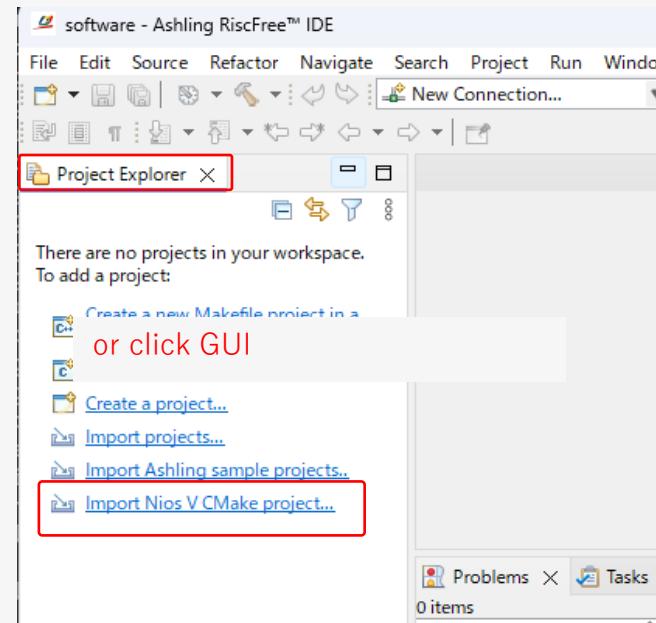
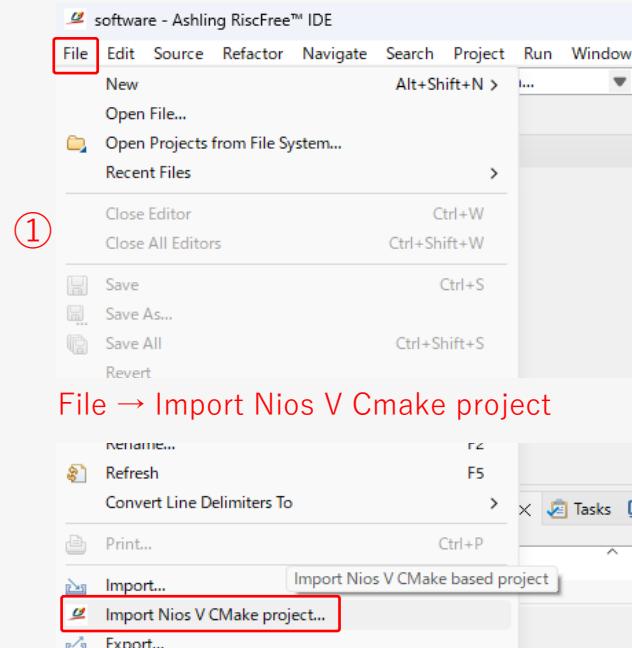
② Generating CMakeLists.txt	
Commands	cmake.exe -G "Unix Makefiles" -DCMAKE_BUILD_TYPE=Debug -B < build execution folder directory > -S < source code directory >
Execution example	cmake.exe -G "Unix Makefiles" -DCMAKE_BUILD_TYPE=Debug -B software/app/build/Debug -S software/app

```
[niosv-shell] C:\$work\%nios5_basic_lab> cmake.exe -G "Unix Makefiles" -DCMAKE_BUILD_TYPE=Debug -B software/app/build/Debug -S software/app  
-- The ASM compiler identification is GNU  
-- Found assembler: C:/intelFPGA/22.1std/riscfree/toolchain/riscv32-unknown-elf/bin/riscv32-unknown-elf-gcc.exe  
-- The C compiler identification is GNU 11.2.0  
-- Detecting C compiler ABI info  
-- Detecting C compiler ABI info - done  
-- Check for working C compiler: C:/intelFPGA/22.1std/riscfree/toolchain/riscv32-unknown-elf/bin/riscv32-unknown-elf-gcc.exe - skipped  
-- Detecting C compile features  
-- Detecting C compile features - done  
-- The CXX compiler identification is GNU 11.2.0  
-- Detecting CXX compiler ABI info  
-- Detecting CXX compiler ABI info - done  
-- Check for working CXX compiler: C:/intelFPGA/22.1std/riscfree/toolchain/riscv32-unknown-elf/bin/riscv32-unknown-elf-g++.exe - skipped  
-- Detecting CXX compile features  
-- Detecting CXX compile features - done  
-- Configuring done  
-- Generating done  
-- Build files have been written to: C:/work/nios5_basic_lab/software/app/build/Debug
```



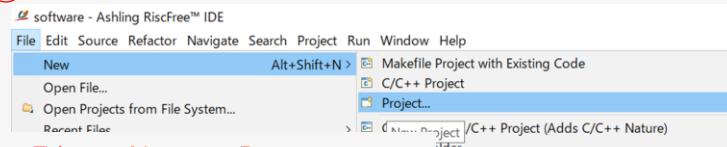
Ashring Importing to RiscFree IDE

- There are two import methods:
 1. Using RiscFree IDE's Import Nios V CMake project
 2. Create New Project
- Import method (1) For Import Nios V CMake project

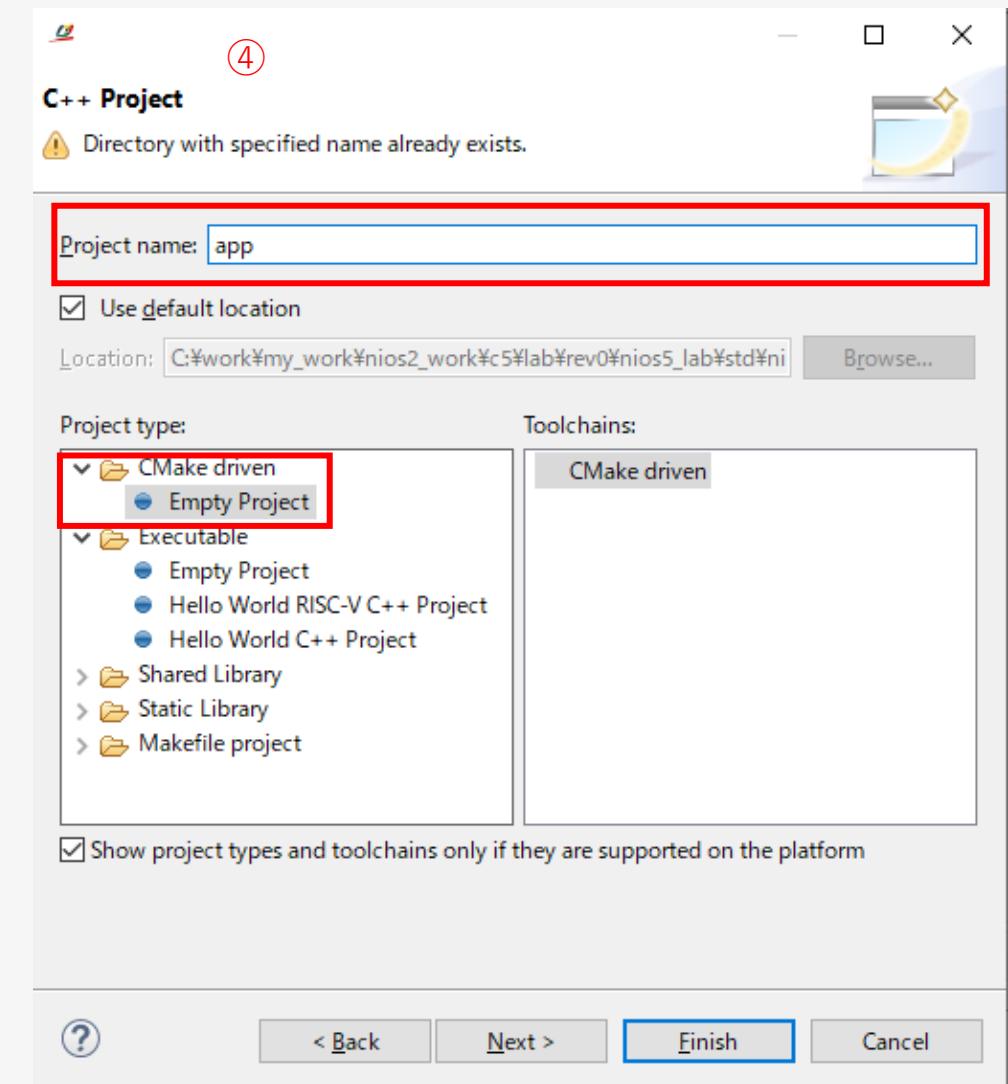
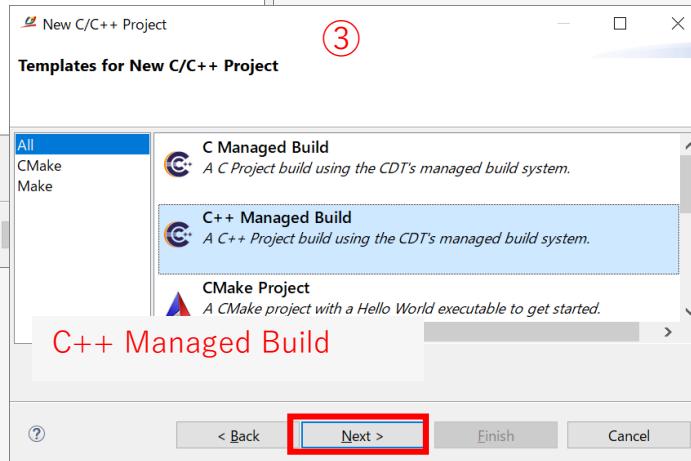
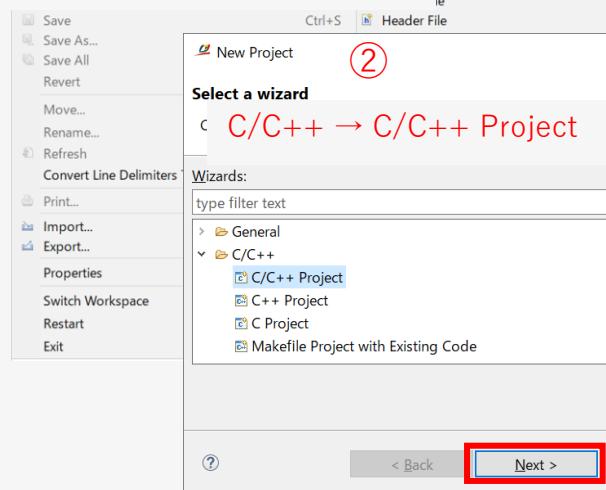


Importing into Ashring RiscFree IDE (for New Project)

①

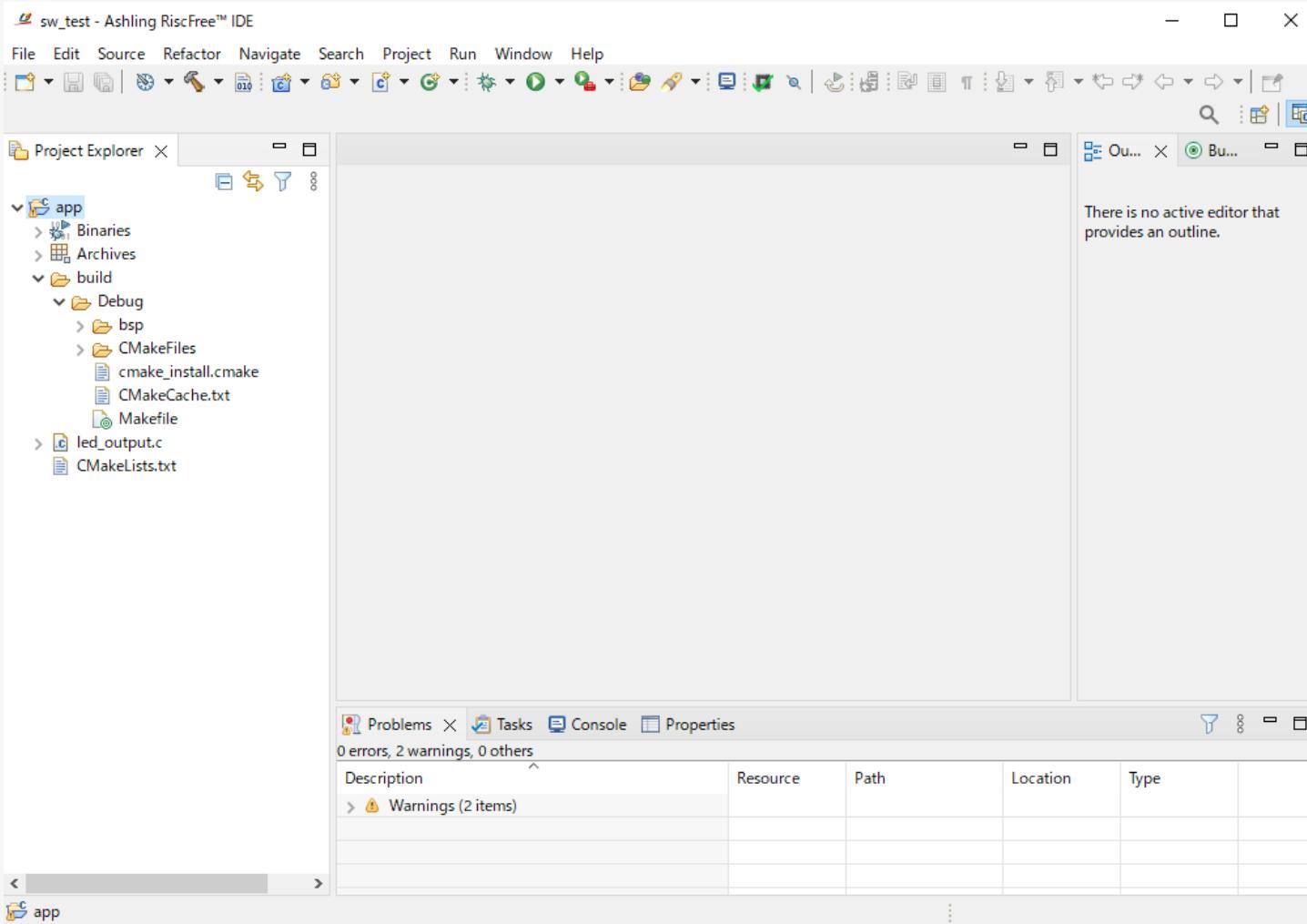


File → New → Project



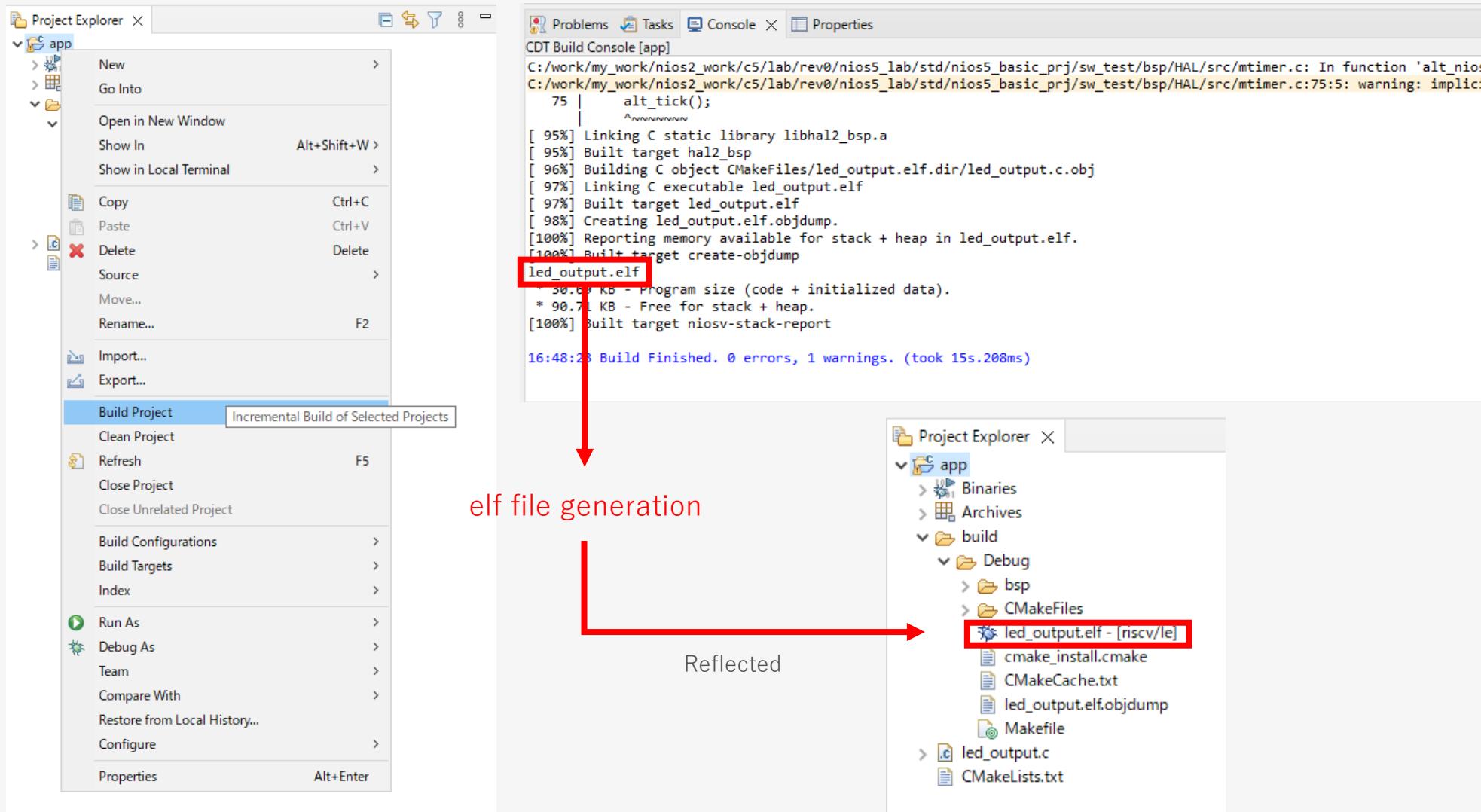
Importing to Ashring RiscFree IDE

- app project appears in Project Explorer after project import completes



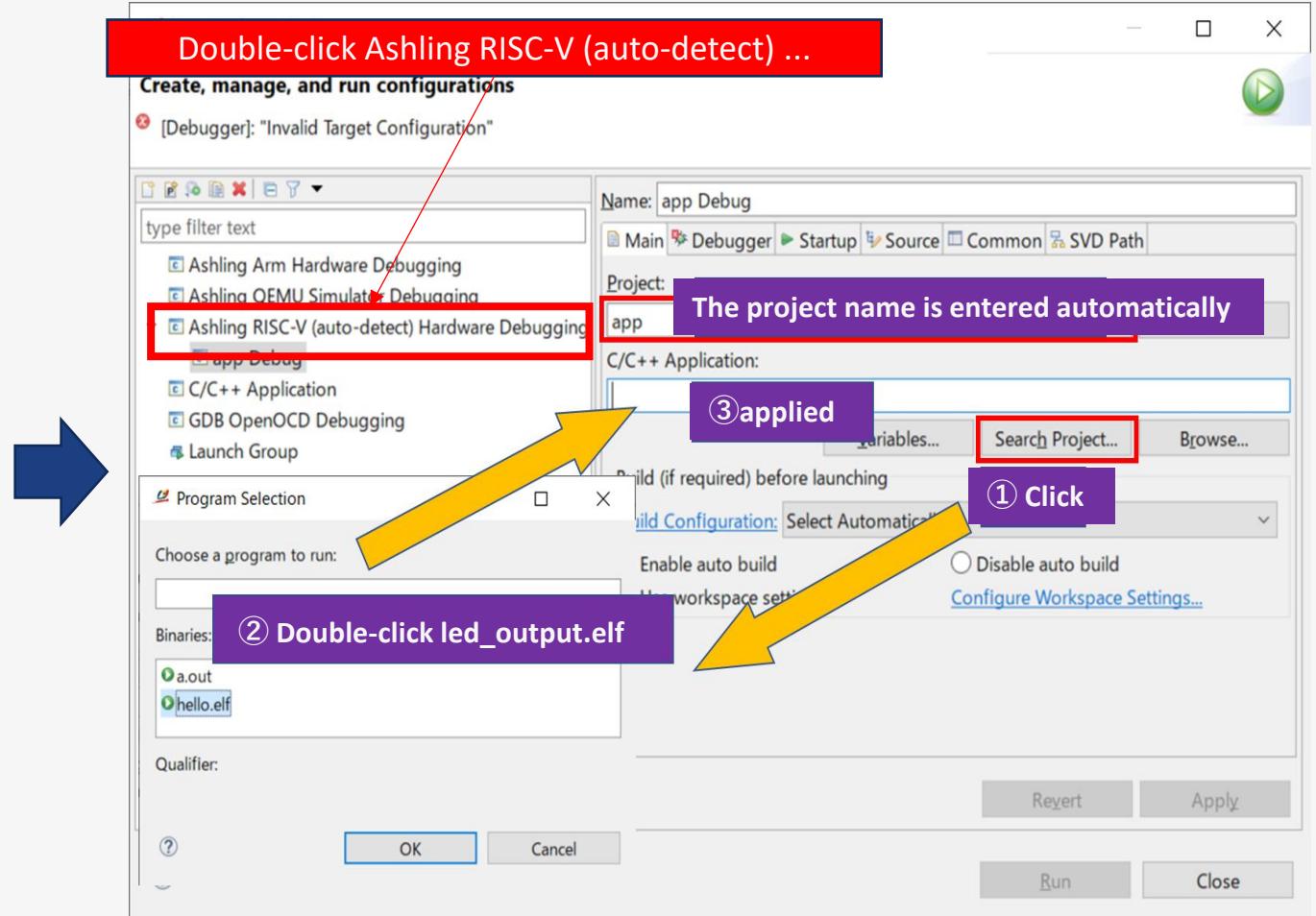
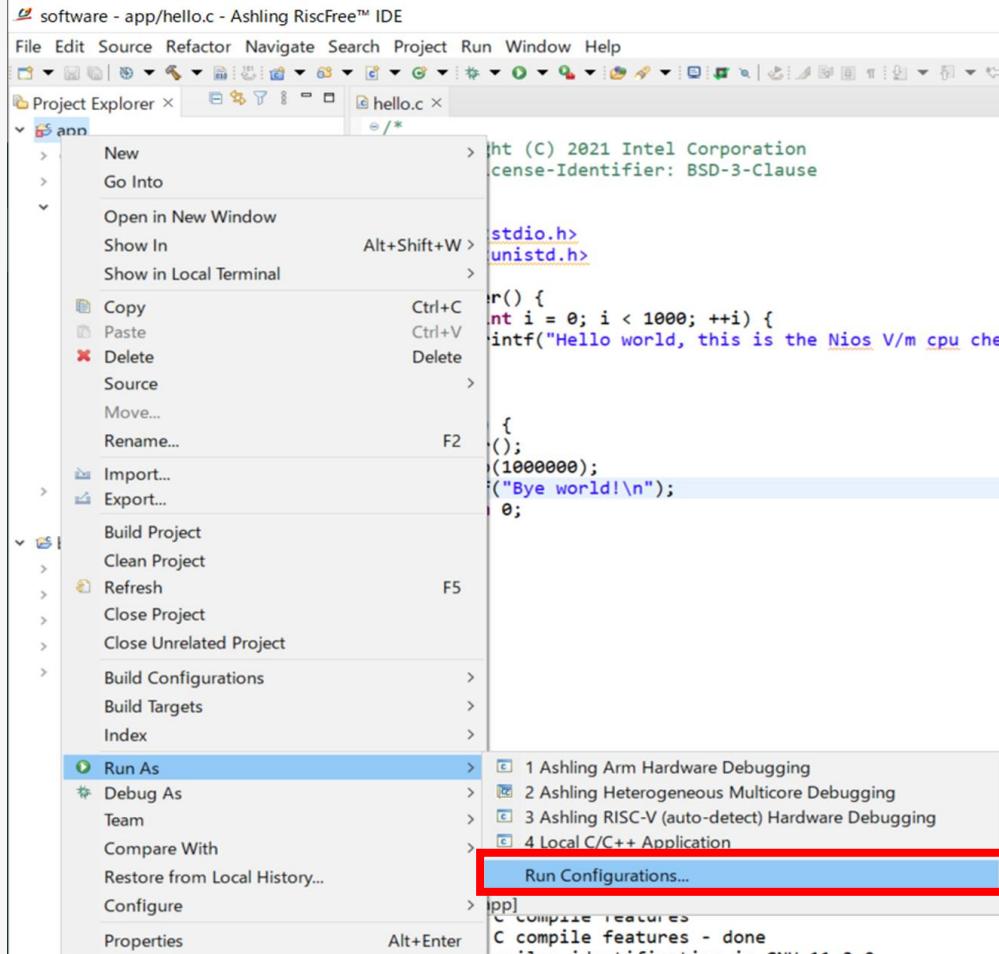
Build the project

- Right-click the app project --> Select Build Project



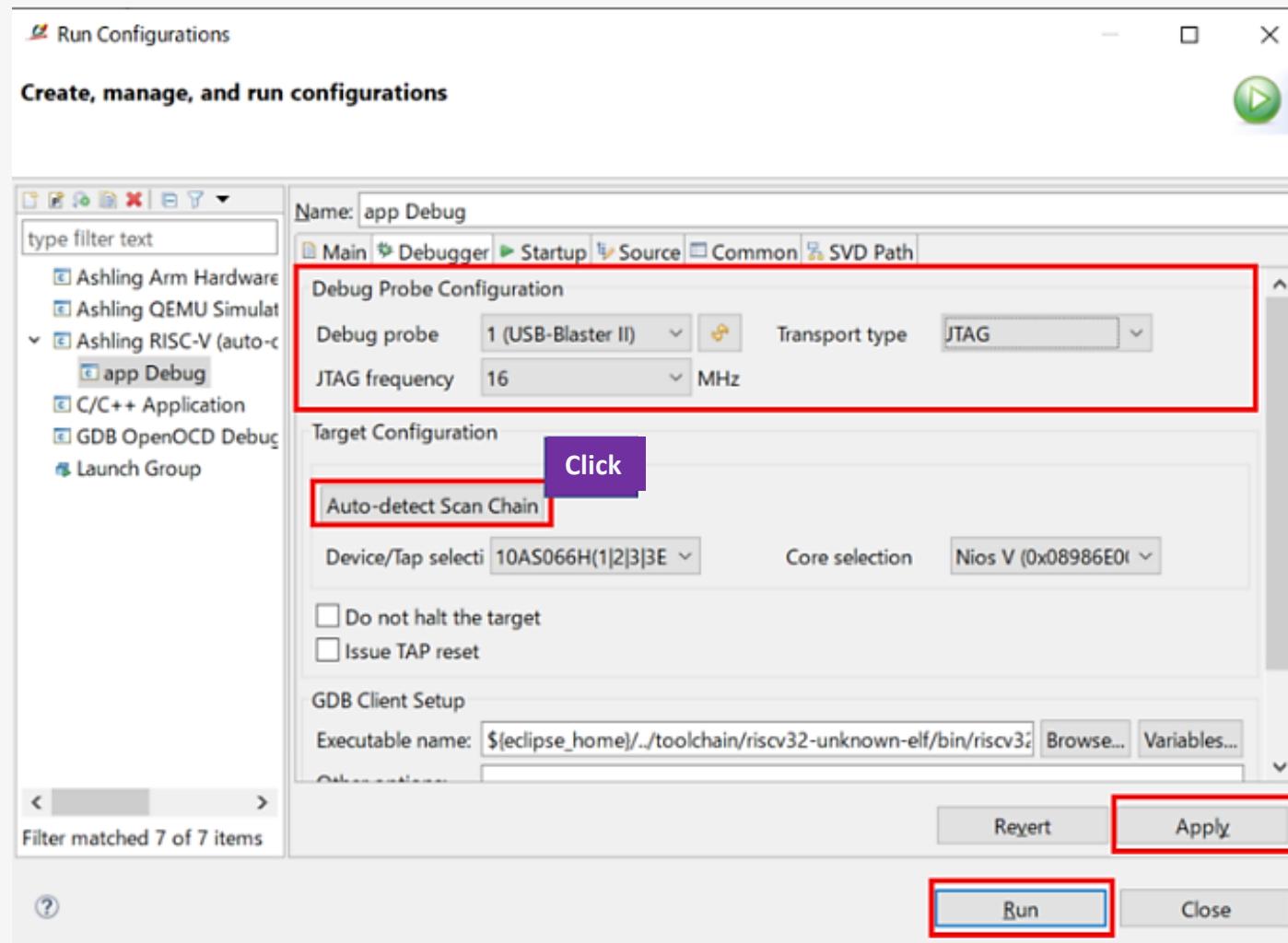
Run the software

- Right-click the app project and select Run As -> Run Configurations



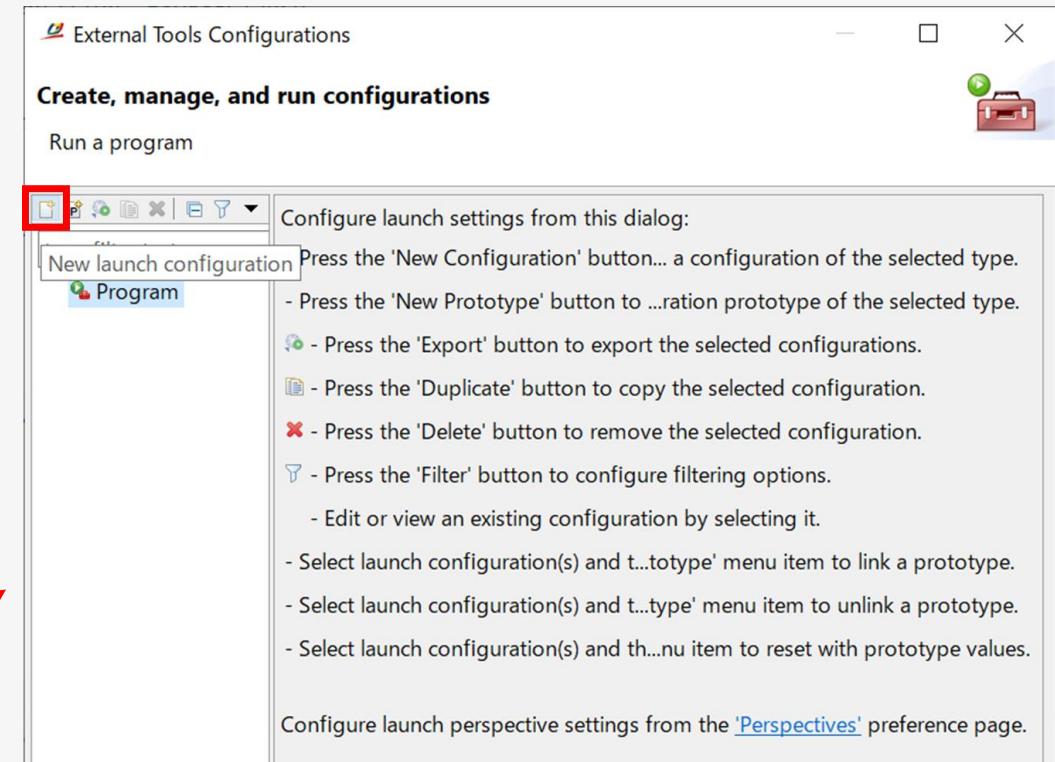
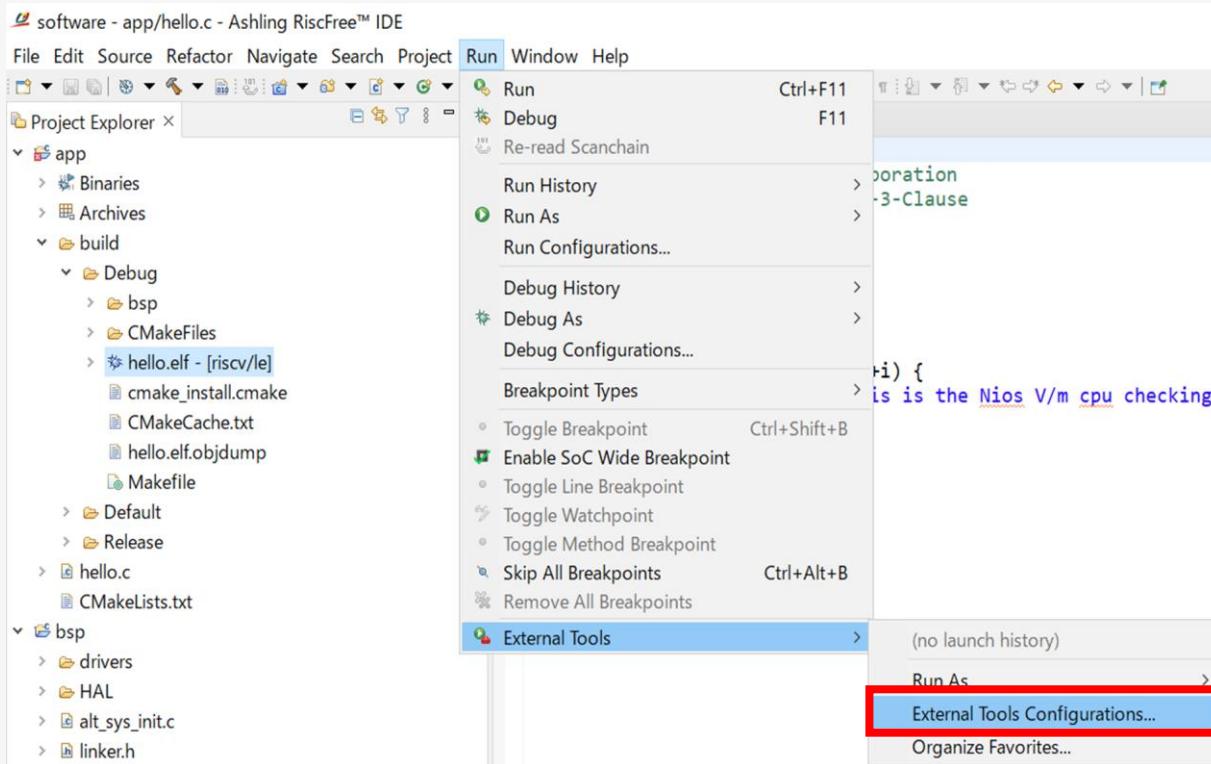
Running the Software

- Open the Debugger tab to configure debug probes and run auto-detect.



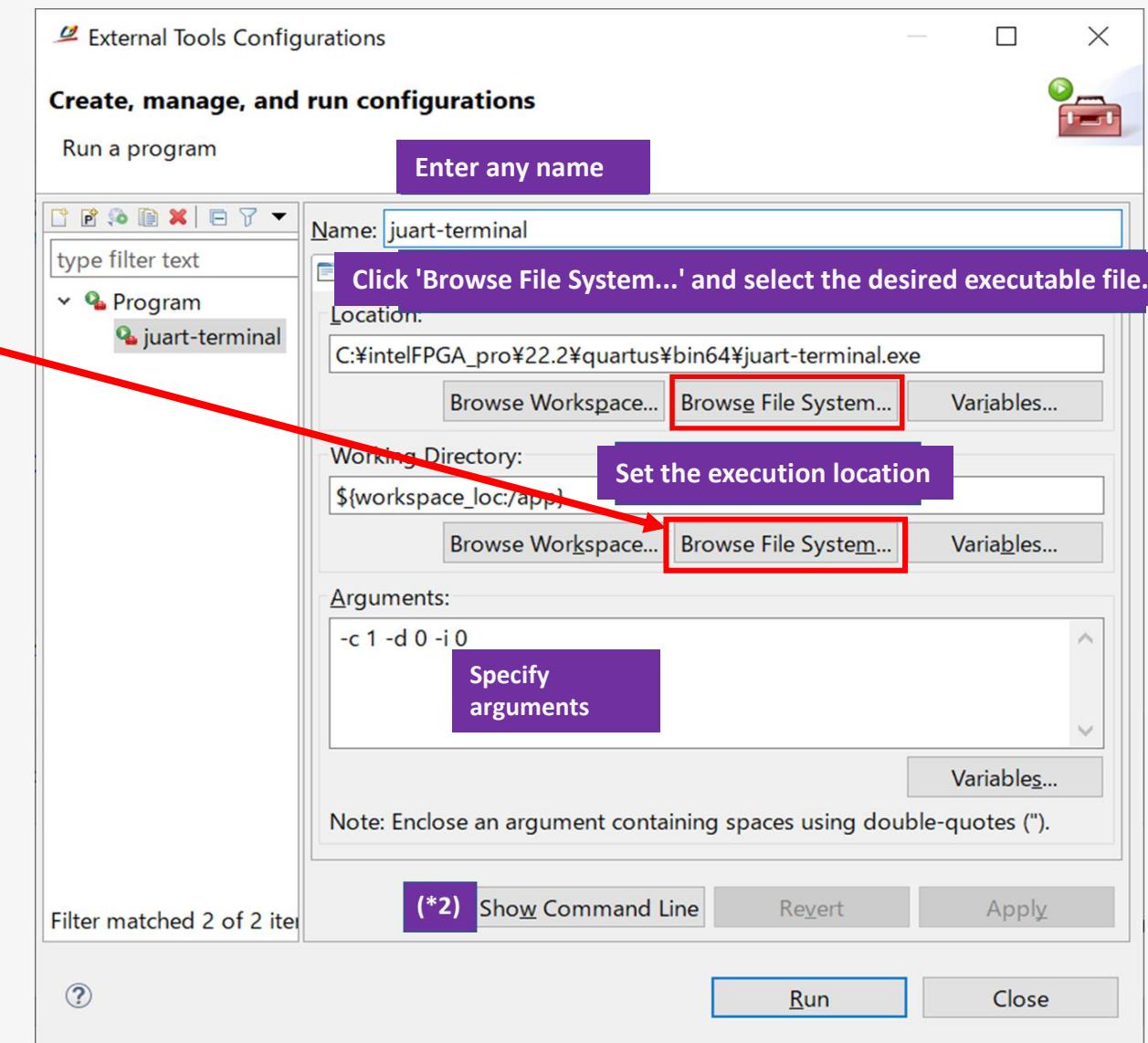
Starting External Tools in the RiscFree IDE

- To register external tools:
 - Run ⇒ External Tools ⇒ Select External Tools Configuration

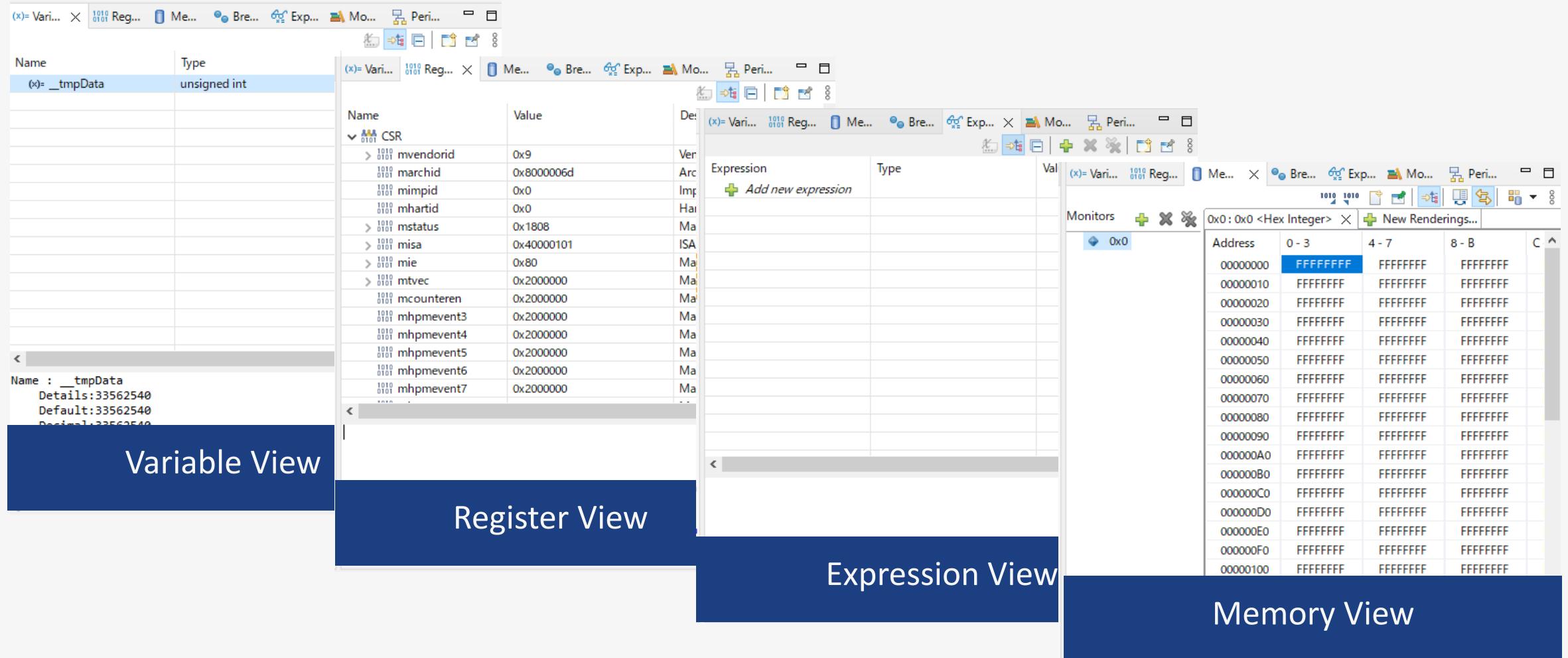


Launching External Tools in the RiscFree IDE

- Register juart-terminal.exe
 - Set Location
 - <quartus DIR>/bin64/juart-terminal.exe
 - Working Directory
 - Select Browse File System



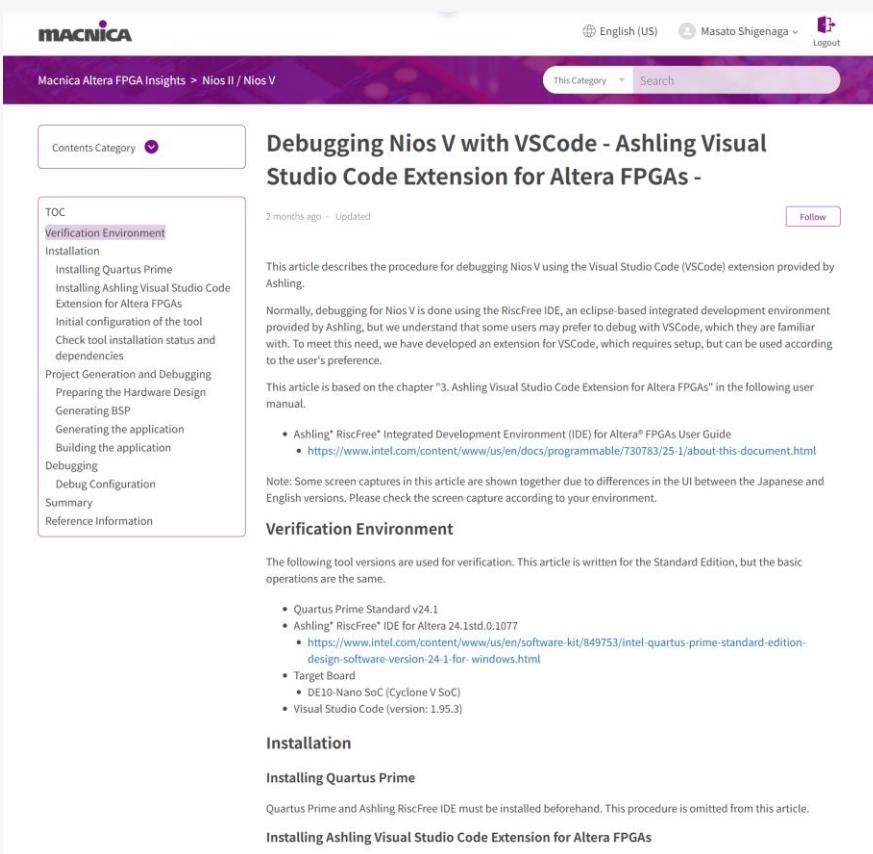
Various Views in the RiscFree IDE



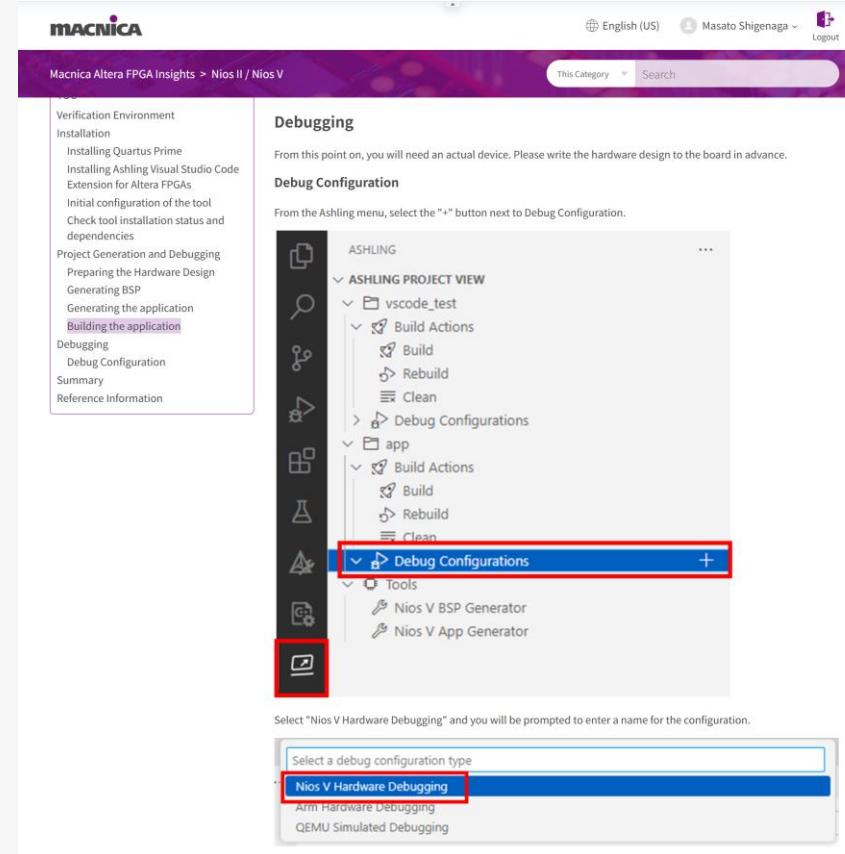
Easy-to-use GUI with Eclipse-based debugging features

Debugging Nios® V with Visual Studio Code (VSCode)

- Ashling provides an extension for Visual Studio Code (VSCode)
- Refer to the following article for how to install and use the tool
 - [Debugging Nios V with VSCode - Ashling Visual Studio Code Extension for Altera FPGAs -](#)



The screenshot shows a web page titled "Macnica Altera FPGA Insights > Nios II / Nios V". The main content is an article with the title "Debugging Nios V with VSCode - Ashling Visual Studio Code Extension for Altera FPGAs -". The article discusses the procedure for debugging Nios V using the VSCode extension provided by Ashling. It includes sections on Verification Environment Installation, Project Generation and Debugging, and Installation.



The screenshot shows the "ASHLING PROJECT VIEW" in the VSCode sidebar. The "Debug Configuration" section is highlighted with a red box. A sub-menu for "Debug Configurations" is open, showing options like "Build", "Rebuild", "Clean", and "Nios V Hardware Debugging". The "Nios V Hardware Debugging" option is also highlighted with a red box.

Agenda

1. Agilex™ 3 FPGA & SoC FPGA Overview
2. What is Nios® V?
3. Migrating from Nios® II
4. Nios® V Hardware Development
5. Exercise 1: Hardware Exercise
6. Nios® V Software Development
7. Nios® V Boot Method
8. Introducing Debugging Tools
9. Exercise 2: Software Exercise
10. What is the FPGA AI Suite?
11. Introduction of Materials
12. Summary

Boot Method

- There are three boot methods for Nios® V.

Boot Method	Overview	Features	
Boot Copier	Expands from non-volatile memory to RAM for execution	<ul style="list-style-type: none"> Faster execution on Cache mishit compared to XIP. Flexibility by editing Boot Copier. Application Requires RAM space to expand code. 	
Execute in Place (XIP)	Runs in non-volatile memory	<ul style="list-style-type: none"> Enables instant boot. Saves RAM space. Slower execution speed on cache mishit compared to Boot Copier. 	
SDM Boot	Deploys to RAM via Secure Device Manager	<ul style="list-style-type: none"> Supported on SDM-equipped devices such as Stratix® 10 and AgilexTM 7. Requires RAM for Bootloader and RAM for Application. Because you need to deploy it to RAM, it works the same as a boot copier. 	

Boot Option types and links to information

Boot method	Boot Memory	Application Code storage location (boot source)	Application Runtime location	Boot Copier	User Guide Link
GSFI Boot (Boot Copier)	QSPI Flash Memory (AS Configuration)	QSPI Flash Memory	RAM (Internal, External)	GSFI bootloader	User Guide
GSFI Boot (XIP)	QSPI Flash Memory (AS Configuration)	QSPI Flash Memory	QSPI Flash Memory	alt_load() function	User Guide
SDM Boot	QSPI Flash Memory (AS Configuration)	QSPI Flash Memory	RAM (Internal, External)	SDM bootloader	User Guide
On Chip RAM Boot	On-Chip Memory (OCRAM)	On chip RAM	On chip RAM	Not required	User Guide
TCM Boot	Tightly Coupled Memory (TCM)	Instruction TCM	Instruction TCM	Not required	User Guide

● Reference

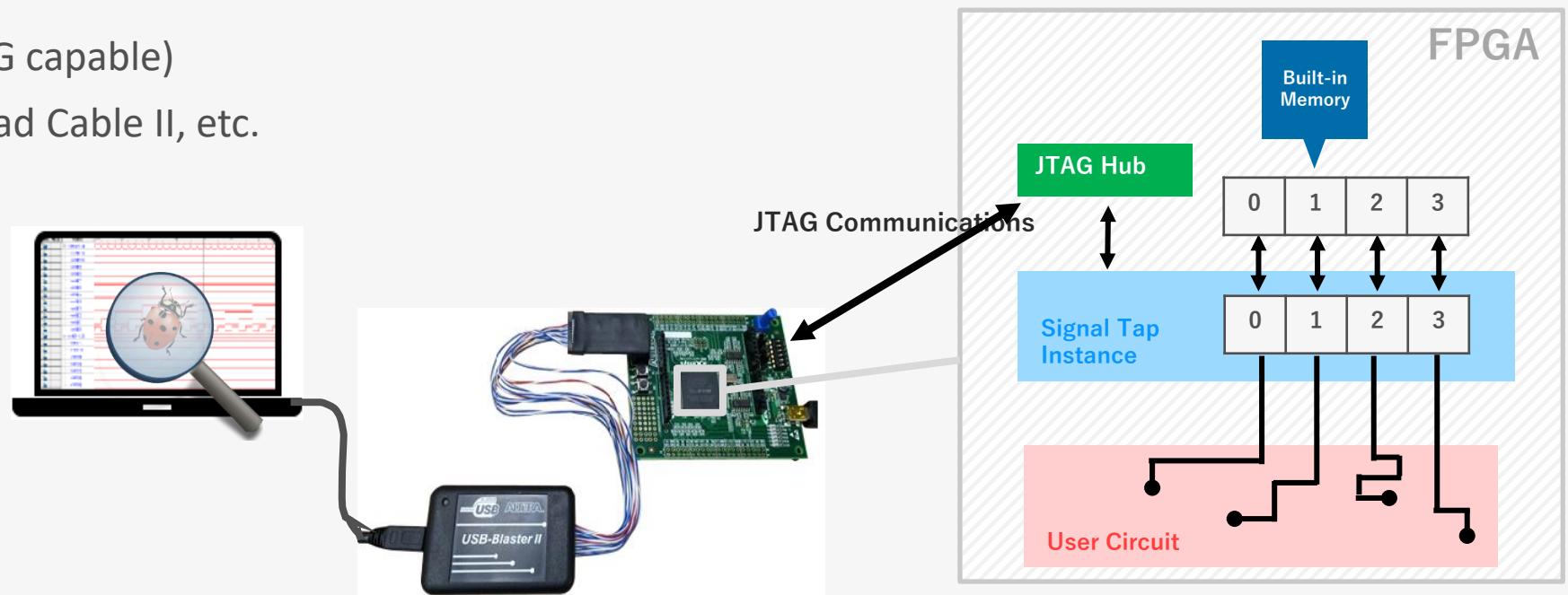
[Nios® V Boot Configuration and Boot Option Settings – Macnica Altima Co., Ltd. \(zendesk.com\)](#)

Agenda

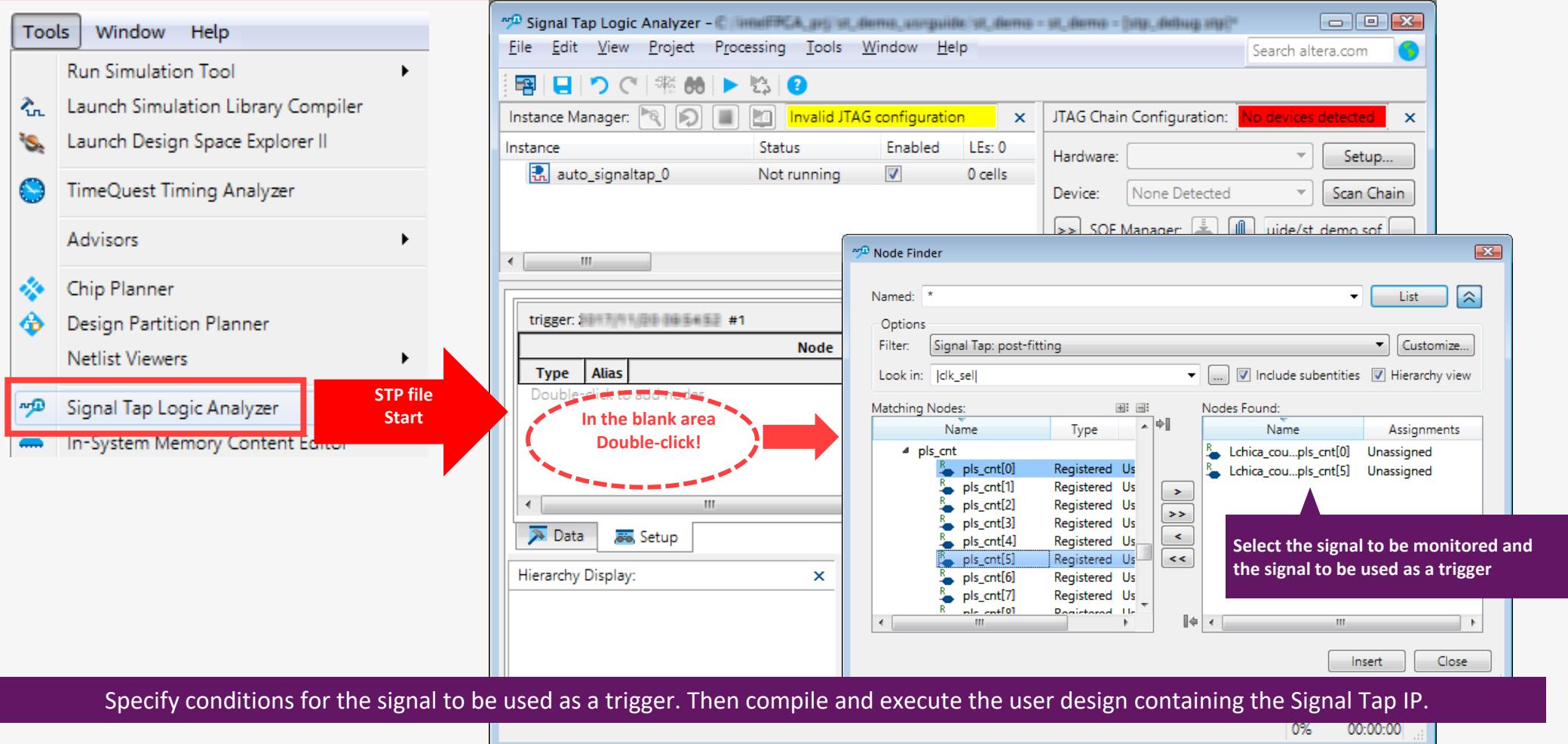
1. Agilex™ 3 FPGA & SoC FPGA Overview
2. What is Nios® V?
3. Migrating from Nios® II
4. Nios® V Hardware Development
5. Exercise 1: Hardware Exercise
6. Nios® V Software Development
7. Nios® V Boot Method
8. Introducing Debugging Tools
9. Exercise 2: Software Exercise
10. What is the FPGA AI Suite?
11. Introduction of Materials
12. Summary

Signal Tap Logic Analyzer

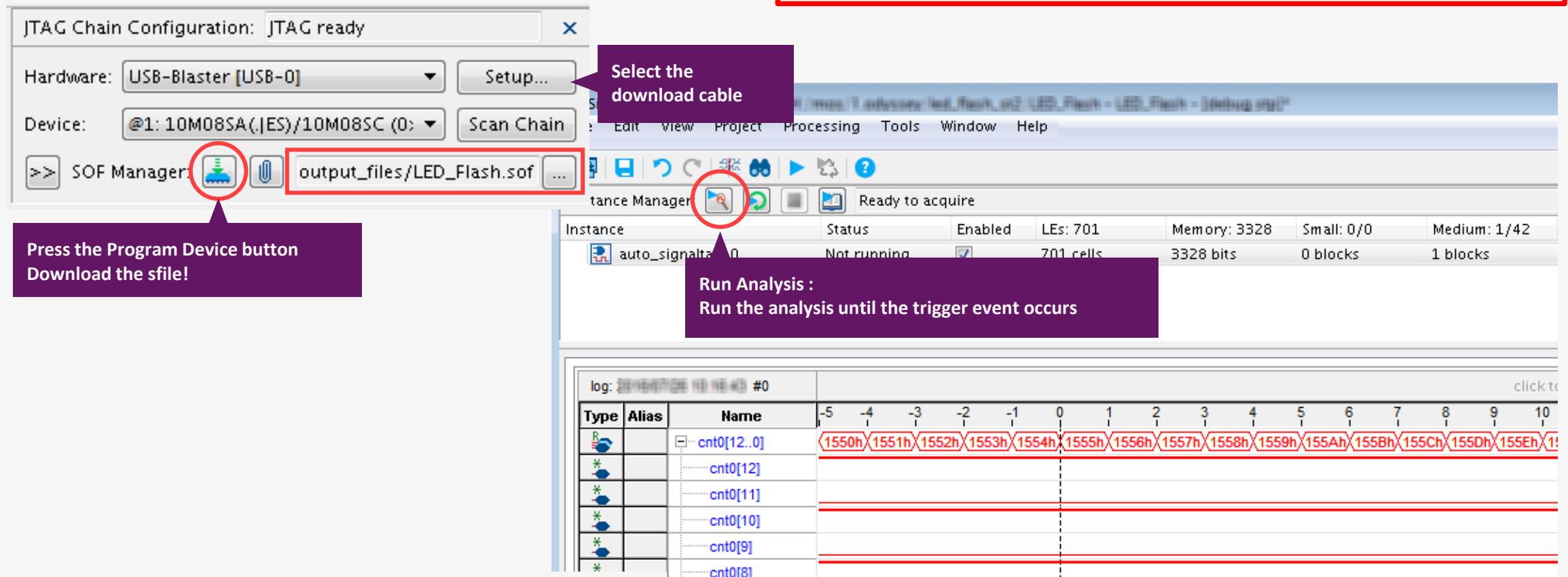
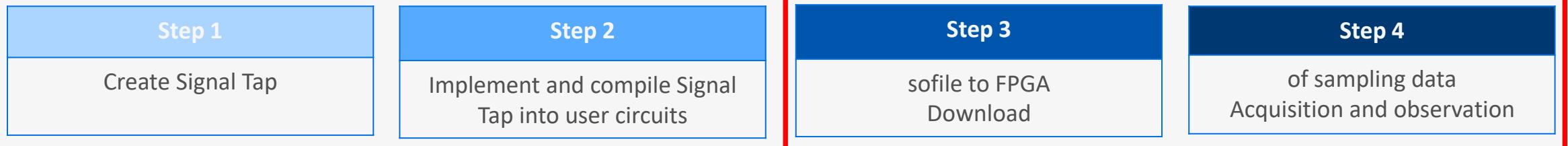
- On-chip debugging method to monitor the status of FPGA internal signals during operation via JTAG
- Integrate the Signal Tap Logic Analyzer IP into the user circuit
 - Signal Tap IP is not licensed (free)
 - Also available in Quartus® Prime Lite Edition
- Monitor I/O pins are JTAG pins only!
- Items required
 - Quartus® Prime
 - Board with FPGA (JTAG capable)
 - Altera® FPGA Download Cable II, etc.



Signal Tap Logic Analyzer Workflow (1/2)



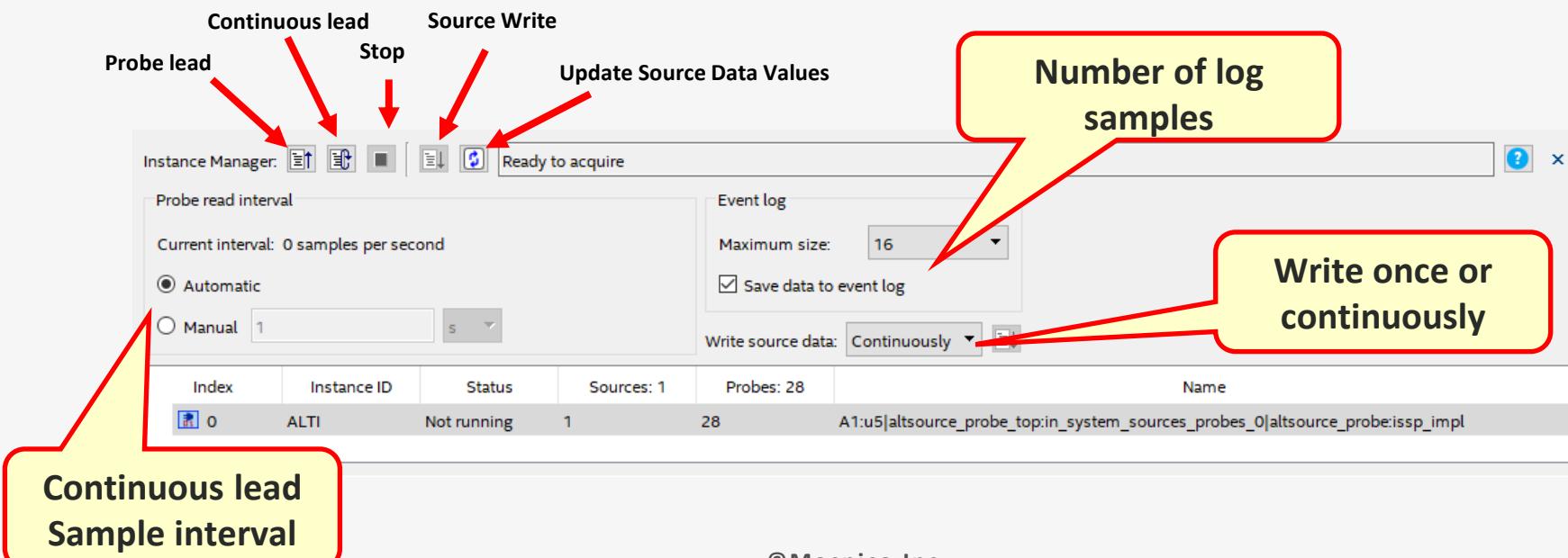
Signal Tap Logic Analyzer Workflow (2/2)



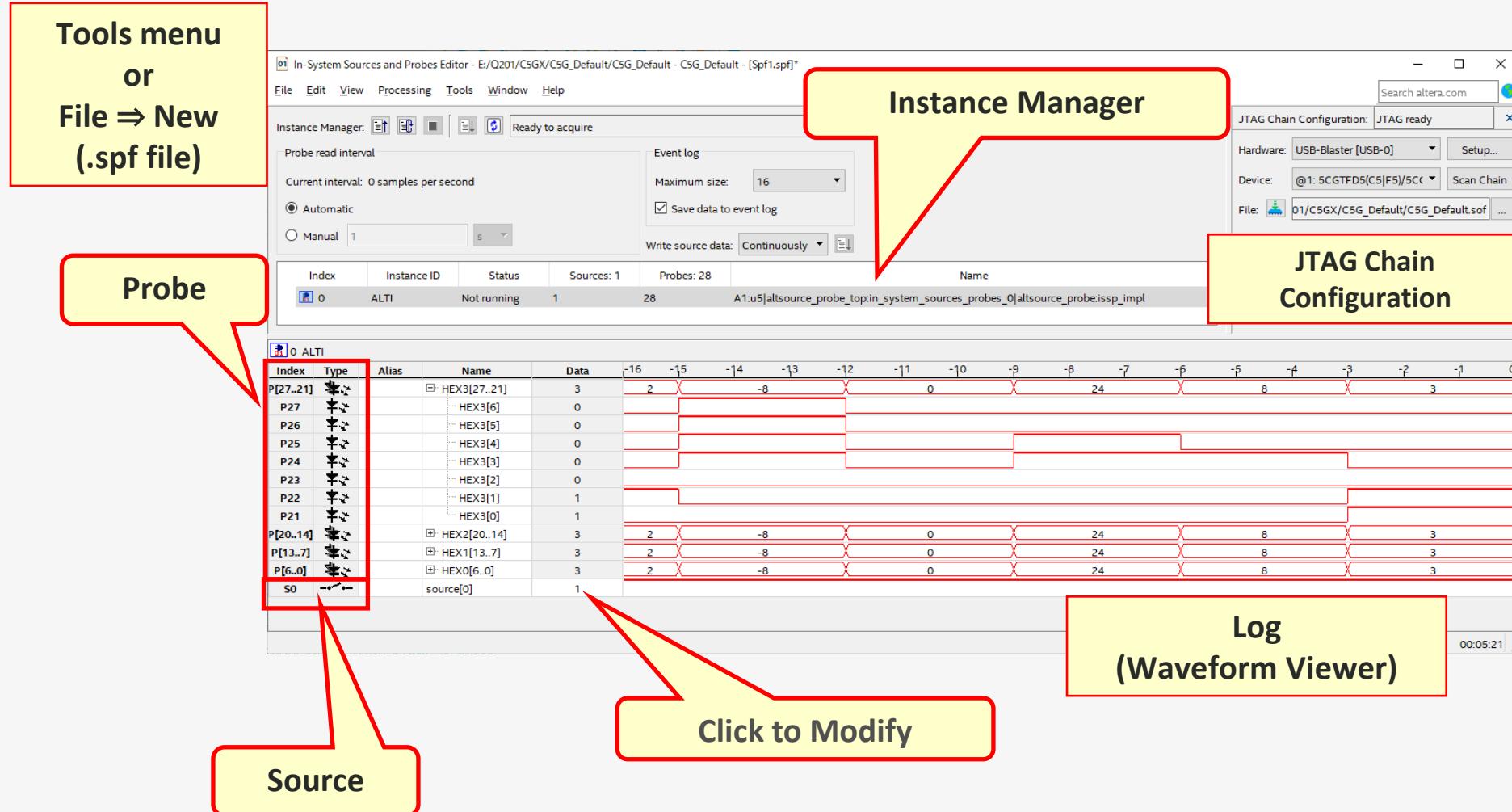
Let's try Agilex™ 3 FPGA Debug "Signal Tap Logic Analyzer"
<https://malt.zendesk.com/hc/articles/49591414544281>

In-System Sources & Probes

- Features
 - Up to 512 signals can be observed while the system is running via JTAG
 - Up to 512 signal values can be driven and toggled in the design via JTAG
- Application
 - Virtual push-button signal control in the design
 - Vary design constants and monitor results
- See web content for details
 - [Let's Try Agilex™ 3 - Basic FPGA functions and usage that you can learn right away](#)

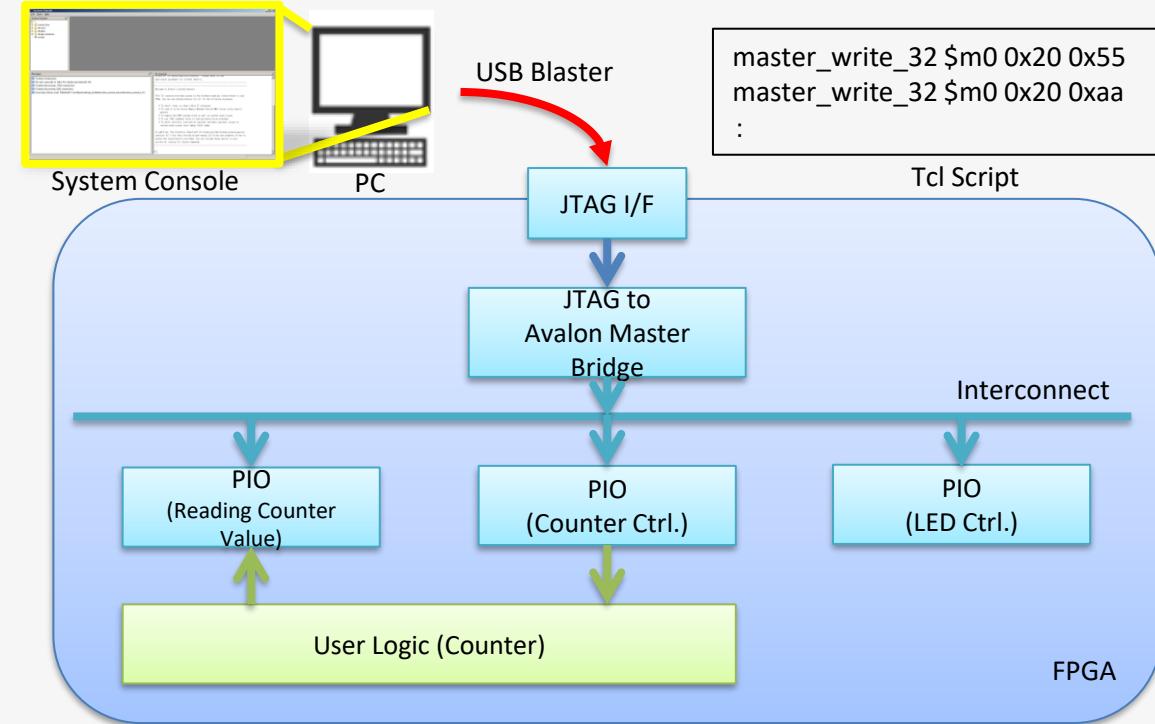


In-System Sources & Probes Editor



System Console

- Features
 - Access resources inside the platform designer via JTAG
 - Easy hardware debugging at board startup
 - Registers can be accessed without creating software
 - Simple GUI can also be generated
 - Can also be used in conjunction with Signal Tap
- Application
 - Debugging at board startup
 - Building a demo environment using a simple GUI
- See the web content for details
 - [Let's Try Agilex™ 3 - Basic FPGA functions and usage that you can learn right away](#)

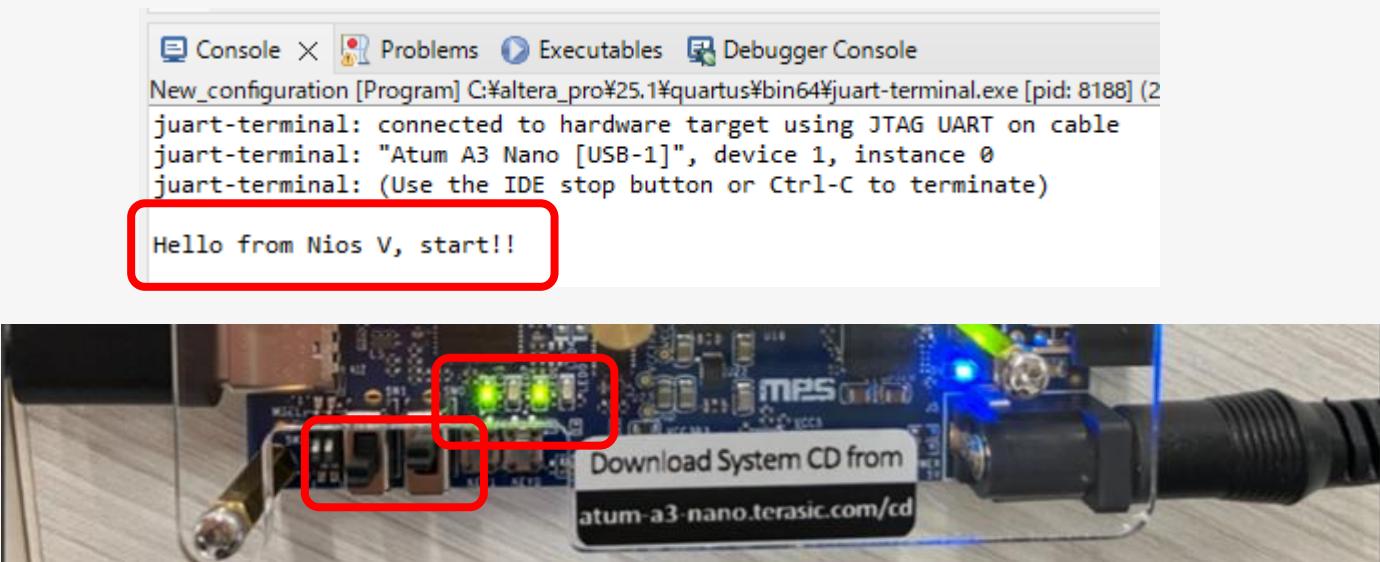


Agenda

1. Agilex™ 3 FPGA & SoC FPGA Overview
2. What is Nios® V?
3. Migrating from Nios® II
4. Nios® V Hardware Development
5. Exercise 1: Hardware Exercise
6. Nios® V Software Development
7. Nios® V Boot Method
8. Introducing Debugging Tools
9. Exercise 2: Software Exercise
10. What is the FPGA AI Suite?
11. Introduction of Materials
12. Summary

Exercise 2: Software Exercise

- Execute the software on the hardware created in Exercise 1
 - Hello from Nios V, start!! Display
 - Read the DIP SW value
 - LED flashing
 - 0x55 ON, 0xAA ON, ALL ON, ALL OFF
 - Flashing interval changes depending on the DIP SW status

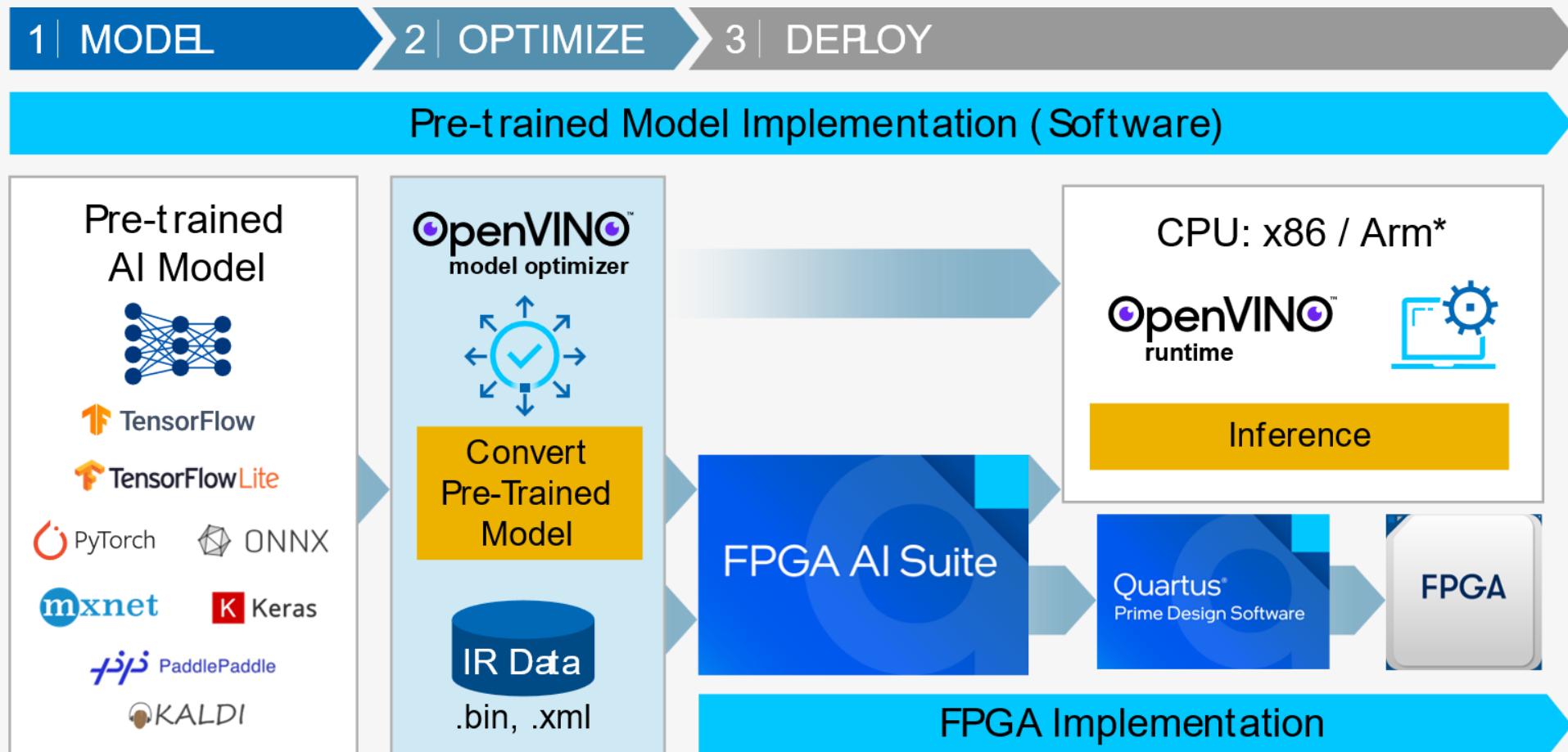


```
C led_output.c x
C: > lab > nios5_lab > pro > nios5_basic_prj > software > app > C led_output.c
1 #include <stdio.h>
2 #include <io.h>
3 #include <unistd.h>
4 #include "system.h"
5
6 #define SLOW 500000
7 #define FAST 250000
8
9 int main(void)
10 {
11     int val;
12     int period = 500000;
13
14     printf("Hello from Nios V, start!!\n");
15
16     period = FAST;
17
18     while(1)
19     {
20         val = (IORD(SW_PIO_BASE,0) & 0x1);
21
22         if (val){
23             period = FAST;
24         } else {
25             period = SLOW;
26         }
27
28         IOWR(LED_PIO_BASE, 0, 0x55);
29         usleep(period);
30         IOWR(LED_PIO_BASE, 0, 0xAA);
31         usleep(period);
32         IOWR(LED_PIO_BASE, 0, 0x00);
33         usleep(period);
34         IOWR(LED_PIO_BASE, 0, 0xFF);
35         usleep(period);
36     }
37
38 }
39
```

Agenda

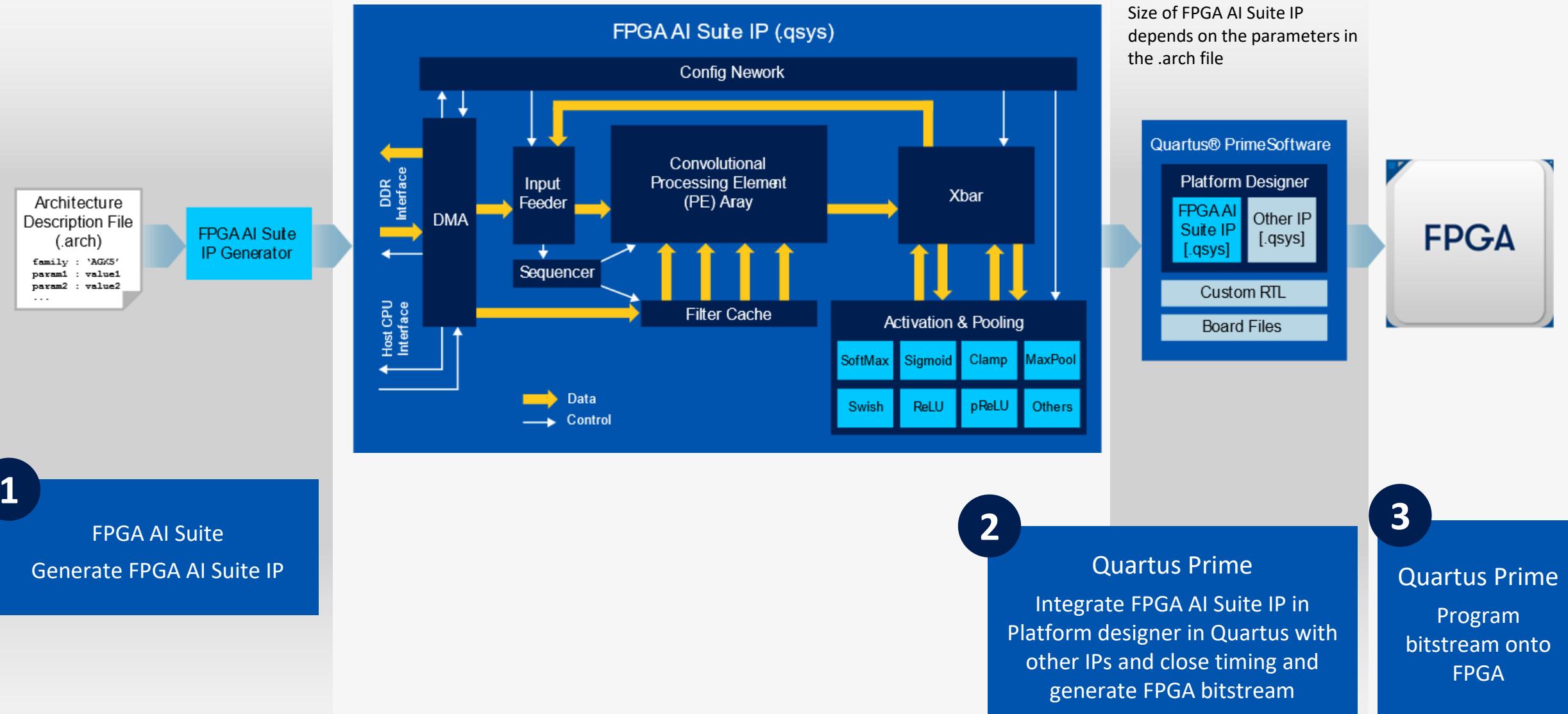
1. Agilex™ 3 FPGA & SoC FPGA Overview
2. What is Nios® V?
3. Migrating from Nios® II
4. Nios® V Hardware Development
5. Exercise 1: Hardware Exercise
6. Nios® V Software Development
7. Nios® V Boot Method
8. Introducing Debugging Tools
9. Exercise 2: Software Exercise
10. What is the FPGA AI Suite?
11. Introduction of Materials
12. Summary

Altera AI – Enabled for Popular Frameworks & Ease of Use



Push-button, AI Inference IP Generation

FPGA AI Inference IP Implementation Flow



Demo

Agenda

1. Agilex™ 3 FPGA & SoC FPGA Overview
2. What is Nios® V?
3. Migrating from Nios® II
4. Nios® V Hardware Development
5. Exercise 1: Hardware Exercise
6. Nios® V Software Development
7. Nios® V Boot Method
8. Introducing Debugging Tools
9. Exercise 2: Software Exercise
10. What is the FPGA AI Suite?
11. Introduction of Materials
12. Summary

Documentation (Agilex™ 3)

- Agilex™ 3 Web Page
 - [Agilex™ 3 FPGA and SoC FPGA Overview](#)
- Agilex™ 3 Documentation
 - [Agilex™ 3 FPGA and SoC FPGA Product and Support Documentation | Altera](#)
 - [Agilex™ 3 FPGAs and SoCs Product Brief](#)
 - [Agilex™ 3 FPGAs and SoCs Device Overview](#)
 - [Agilex™ 3 FPGAs and SoCs Device Data Sheet](#)
- Macnica Materials
 - [Let's Try Agilex™ 3 - Basic FPGA functions and usage that you can learn right away](#)



Documentation (Nios® V)

● Nios® V Web Page

- [Nios® V Processor Family](#)

● Documentation

- [Nios® V Processor Reference Manual](#)
- [Nios® V Processor Embedded Processor Design Handbook](#)
- [Nios® V Processor Software Developer's Handbook](#)
- [Nios® V Processor RiscFree IDE for Intel® FPGAs](#)
- [AN978: Nios® V Processor Migration Guide](#)
- [AN985: Nios® V Processor Tutorial](#)

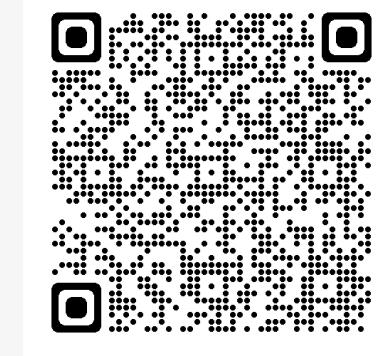
● Quick Videos

- [Nios® V Processor for Intel® FPGAs \(intel.com\)](#)
- [Nios® V Processor Design Walkthrough Video](#)
- [Design Upgrade from Nios® II V to Nios® V Video](#)

● Macnica Materials

- [Nios® V Processor - Intel - Macnica](#)
- [Nios V Summary Page](#)
- [Nios® V Project Development Procedure Using Ashling® RiscFree® IDE](#)
- [How to Obtain a License File for Nios® V Processor IP](#)
- [Configuration of Booting Nios® V and settings for each Boot Option](#)

Intel
Introduction Page



Macnica
Introduction Page



Nios® V Processor

Nios® processors are the next generation of soft processor IPs, designed to bring the power and flexibility of the open-source RISC-V architecture to FPGA environments. By leveraging the RISC-V instruction set architecture (ISA), the Nios V processors offer valuable solutions that enable a spectrum of applications ranging from simple embedded systems to complex, high-performance applications.

Altera has discontinued the Nios® II IP ordering codes. Refer to PSIN 2312 for more information. It is recommended that customers migrate to Nios V.

[Read the Nios® V Processor reference manual](#)

[Overview](#) [Documentation](#) [Ecosystems](#) [What's New](#) [Videos](#) [Embedded Software](#)

Key Benefits

Open-Source RISC-V Community

- Leverage the power of the community-maintained ecosystem to expedite your path to market.
- Develop your designs using debuggers, and test-and-debug environments (IDEs) available from the RISC-V ecosystem.
- Utilize the Nios® V Processor IP cores to design your own processor or integrate existing Nios II designs.
- Design and integrate with pre-existing IP cores within Platform Designer.

Simple and Standard Flows with Easy System Integration

- Utilize traditional hardware tool flows, such as Platform Designer and the Quartus® Prime Software.
- Implement the Nios® V Processor IP core in your design.
- Debug your design with the supported Joint Test Action Group (JTAG) debug module, offering on-chip debugging capabilities.

Flexibility and High-Performance

- Real high performance levels, and unparalleled flexibility for your embedded design with the Nios V processor:
- Choose from three different Nios® V Processor IP core frequencies and power-hungry off-the-shelf processors.
- Integrate multiple CPUs, peripherals, memory interfaces, and custom peripherals.

Three Nios® V Processors to Meet Your Design Requirements

[Watch the Nios V Processor Portfolio video](#)

[Nios® V Processor for Intel® FPGAs](#)

FPGAに組み込む
オリジナル・プロセッサー

Nios® V (223.7MHz, 704KB SRAM, 4.77GHz) Nios® (323.7MHz, #cores=7+8+9+8+11),
Nios® FPGA (323.7MHz, 27.7MHz)

– Nios® V の特徴

Nios® V の特徴 - フィーチャーと機能
Nios® V はRISC-Vアーキテクチャ
Nios® Vは、アルゴリズムを実行するための柔軟性と並列性を備えています
Nios® Vは、複数のCPUを統合して複数の機能を実現するための機能を備えています
Nios® Vは、複数のIPコアを統合して複数の機能を実現するための機能を備えています
Nios® Vは、複数のIPコアを統合して複数の機能を実現するための機能を備えています
– Nios® V ファミリー

Nios® V ファミリー - 製品選択
Nios® V ファミリーは、Nios® V (4.77GHz), Nios® V (3.23GHz), Nios® V (2.23GHz) の3つのモデルがあります
Nios® V (4.77GHz)は、高機能なプロセッサーで、複数のCPUを統合して複数の機能を実現するための機能を備えています
Nios® V (3.23GHz)は、中程度の性能で、複数のIPコアを統合して複数の機能を実現するための機能を備えています
Nios® V (2.23GHz)は、低機能なプロセッサーで、複数のIPコアを統合して複数の機能を実現するための機能を備えています

[Nios® V Processor - Intel - MacNica \(macnica.co.jp\)](#)

etc.

AN 978: Nios® V Processor Migration Guide

AN 978: Nios® V Processor Migration Guide

[Download ↓](#)[Bookmark ↗](#)[View More ▾](#) Search this document

- 1. Overview
- 2. Architecture Comparison
- 3. Processor Core Migration Mapping
- 4. Design Flow Comparison
- 5. Migration by Use Case
- 6. References
- 7. Document Revision History for AN 978: Nios® V Processor Migration Guide

4. Design Flow Comparison

Table 4. Migration Consideration by Design Flow

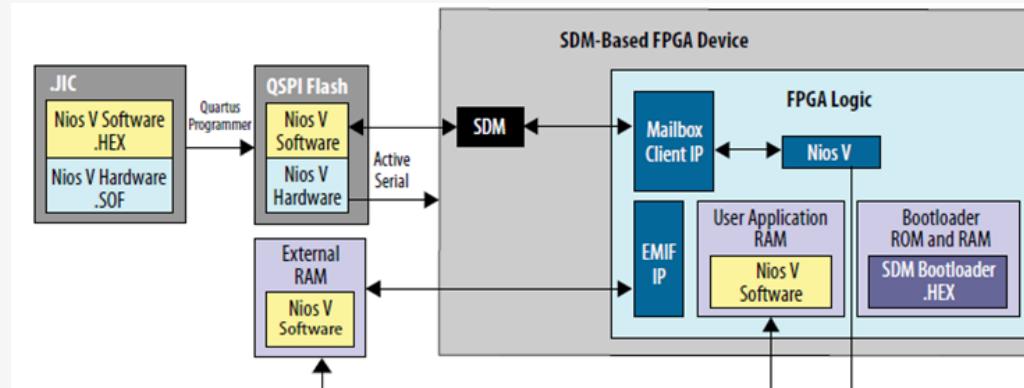
Design Stage	Nios® V Processor	Nios® II Processor	Migration Consideration
Intel® Quartus® Prime Project Creation	Create a new project using the New Project Wizard .	Create a new project using the New Project Wizard .	None
Define and Generate System in the Platform Designer	1. Instantiate the Nios® V processor core. 2. Create a Nios® V processor system with basic peripherals.	1. Instantiate the Nios® II processor core. 2. Create a Nios® II processor system with basic peripherals.	1. Nios® V processor has a similar interface as the Nios® II processor. You can replace the interface in the Platform Designer. 2. Refer to the Table <i>Core Migration</i> for the processor core mapping guidelines. 3. Refer Table <i>Primary Interface</i> for the signals connection mapping guidelines.
Invoke the BSP Editor	<ul style="list-style-type: none">• For Intel® Quartus® Prime Pro Edition software:<ul style="list-style-type: none">◦ Use the Platform Designer.• For Intel® Quartus® Prime Standard Edition software:<ul style="list-style-type: none">◦ Use the Nios® V Command Shell with the command <code>niosv-hsp-</code>	<ul style="list-style-type: none">• You can use the Nios® II Embedded Design Suite (Eclipse IDE) in both Intel® Quartus® Prime software editions.• Alternatively, you can use the following software:<ul style="list-style-type: none">◦ For Intel® Quartus® Prime Pro	Invoke the BSP Editor with a different tool.

<https://www.intel.com/content/www/us/en/docs/programmable/773196/current/design-flow-comparison.html>

Reference Design

- Available from [FPGA Design Store](#) and [Altera FPGA Developer Site](#)
 - [AgilexTM 5 FPGA - Hello World and OCM Memory Test Design Example on Nios® V/m Processor](#)
 - [AgilexTM 5 FPGA - Hello World on Nios® V/g Processor Design Example](#)
 - [AgilexTM 5 FPGA - Nios® V/g Processor OCM to OCM Design Example](#)
 - [AgilexTM 5 FPGA - Nios® V/m PIO LED Toggle Design Example](#)
 - [AgilexTM 5 FPGA - Nios® V/m Processor with DMA and OCM Design Example](#)
 - [AgilexTM 5 FPGA - Drive-On-Chip Design Example](#)
 - [Intel® Stratix® 10 FPGA – SDM Bootloader for the Nios® V/m Processor Design Example](#)
 - [Intel® Arria® 10 FPGA – Simple Socket Server for the Nios® V/g Processor Design Example](#)
 - [Intel® MAX® 10 FPGA - Helloworld on Nios® V/m Processor Design Example](#)

*Above are some of the design offerings



Nios® V License



- License is free

- Must be obtained from Altera® FPGA Self-Service License Center
- Macnica Web Article
 - [How to obtain a license file for Nios® V processor IP](#)

The screenshot shows the Macnica website with the following details:

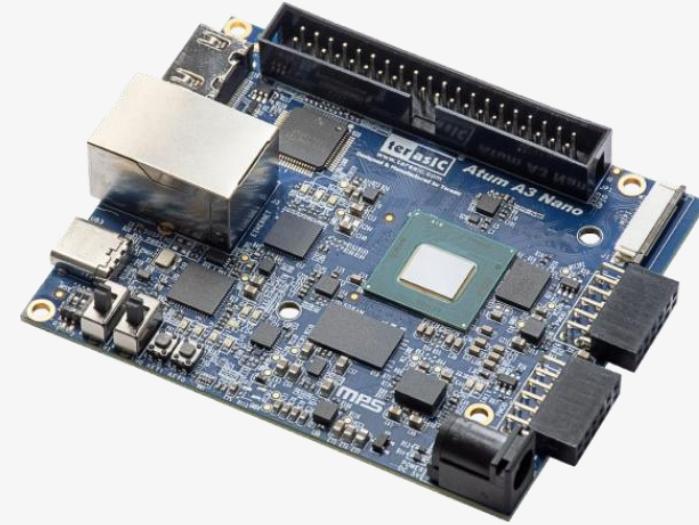
- Article Title:** How to Obtain a License File for Nios® V Processor IP
- Date:** 2024.02.01
- Content Summary:** Link the license to the registered computer.
- Steps:**
 - Click the [Evaluation or free license registration] menu.
 - Select the Nios® V license you want to acquire from the product list, referring to the following:
 - I would like Nios V/c license ⇒ (License: IP-NIOSVC) choose
 - I would like a Nios V/m license ⇒ (License: IP-NIOSVM) choose
 - If you want a Nios V/g license ⇒ Select (License: IP-NIOSVG)
 - Enter the number of licenses for the NIC ID you are applying for in the number of seats field located at the bottom left. Enter 1 for FIXED and the number to be operated for FLOAT.
 - Click the Next button.
- Inset Screenshot:** Shows the Intel® FPGA Self-Service Licensing Center interface with a red box highlighting the "評議版または無制限ライセンスの選択" (Selection of Evaluation or Free License) button. Below it, a red box highlights the "Nios® Vm Microcontroller Intel® FPGA IP (License: IP-NIOSVM)" entry in the list of products.

For the next step

- Guide to the evaluation board



[Agilex™ 3 FPGA C-Series
Development Kit](#)



[Terasic Atum A3 Nano
Development Kit](#)

Offering ready-to-try designs

- Offering designs checked at Macnica on the web
 - [Let's Try Agilex™ 3 - Basic FPGA functions and usage that you can learn right away](#)
 - Expected to expand in the future

The screenshot shows a landing page for the 'Let's Try Agilex™ 3' basic design. It features a large image of the Atum A3 Nano evaluation board. The main title is 'Let's Try Agilex™ 3 - Basic FPGA functions and usage that you can learn right away'. Below the title, there is a brief introduction and a detailed section on 'Introduction' which explains what FPGAs are and how they work. There is also a section on 'Configuration section' and 'Evaluation and measurement'.

The image contains four screenshots of Macnica Altera FPGA Insights articles:

- Let's Try Agilex™ 3 - Debugging FPGA (Signal Tap Logic Analyzer)**: This article covers how to use the Signal Tap Logic Analyzer to debug an FPGA. It includes a TOC with sections like 'Introduction', 'environment setup', 'design description', 'Operation of Signal Tap Logic Analyzer', 'conclusion', and 'attachments'.
- Let's Try Agilex™ 3 - Measuring Internal Voltage Values During Operation**: This article discusses measuring internal voltage values during operation. It includes a TOC with sections like 'Introduction', 'voltage sensor', 'voltage verification procedure (Configuration Debugger)', 'configure the device', 'start Configuration Debugger', 'Hardware Setup and Load Device Settings', 'voltage confirmation', and 'Conclusion'.
- Let's Try Agilex™ 3 - I/O PLL dynamic reconfiguration**: This article explains dynamic reconfiguration of I/O PLLs. It includes a TOC with sections like 'Introduction', 'I/O PLL Types', 'Preparing to Perform Dynamic Reconfiguration', 'dynamic reconfiguration execution procedure', 'checking actual device operation', 'creating a sample design', 'Execution of System Console', 'Checking with Signal Tap II', and 'Conclusion'. It also includes a diagram of the Atum A3 Nano board showing power supply connections.
- [Figure 1] Voltage Sensor (Source: Power Manager)**: This is a technical diagram titled 'Figure 1' showing the 'Voltage Sensor' from the 'Power Manager'. It illustrates the connection between the board's power supplies and the voltage sensor.

LET'S TRY Agilex™ 3



Agenda

1. Agilex™ 3 FPGA & SoC FPGA Overview
2. What is Nios® V?
3. Migrating from Nios® II
4. Nios® V Hardware Development
5. Exercise 1: Hardware Exercise
6. Nios® V Software Development
7. Nios® V Boot Method
8. Introducing Debugging Tools
9. Exercise 2: Software Exercise
10. What is the FPGA AI Suite?
11. Introduction of Materials
12. Summary

Summary

- Agilex™ 3 is the next generation power and cost optimized FPGA
 - Uses Intel 7 process
 - Uses 2nd generation Hyperflex™ FPGA architecture
 - Rich interface
 - Available as of Quartus® Prime Pro v25.1
 - Get a sample to try right away
 - [Let's Try Agilex™ 3 - Basic FPGA functions and usage that you can learn right away – Macnica Altera FPGA Insights](#)
- Nios® V is a next-generation soft-core processor
 - Uses open RISC-V architecture
 - Rich ecosystem



Agilex™ 3/Nios® V is ready to evaluate!
Thank you for your consideration



- The names of companies, products or services mentioned in this document are trademarks or registered trademarks of the respective companies. "TM" and "®" are not specified in this document.
- All rights reserved. No part of this document may be reproduced or duplicated in any form or by any means (except as permitted by copyright law) without the prior written consent of Macnica, Inc. or a third party.
- The information in this document is based on information available as of the date of preparation, but Macnica does not guarantee the accuracy or completeness of the information.