Nuclei® E603 Databook

Nucleisys

Contents

1 Copyright Notice

The product described herein is subject to continuous development and improvement; information herein is given by Nuclei in good faith but without warranties.

This document is intended only to assist the reader in the use of the product. Nuclei do not assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation indirect, incidental, special, exemplary, or consequential damages. incorrect use of the product.

2 Contact Information

Should you have any problems with the information contained herein or any suggestions, please contact Nuclei System Technology by email support@nucleisys.com, or visit "Nuclei User Center" website http://user.nucleisys.com for supports or online discussion.

List of Figures

List of Tables

3 Revision History

Rev.	Revision Date	Revised Section	Revised Content
1.0.0	2025/8/8	N/A	1.Initial version

4 Overview

The Nuclei Hummingbird E603 is a free 64-bit RISC-V application processor that supports Linux, introduced by Nuclei System Technology and specifically designed to meet academic and educational needs.

4.1 Feature List

The Nuclei Hummingbird E603 have the following features:

- CPU Core
  - Nuclei Hummingbird E603: support RV64GC with MMU, for the 64-bit Linux capable applications.
  - 6-pipeline stages, using state-of-the-art processor micro-architecture to deliver the best-of-class performance efficiency and lowest cost.
- Multiplier and Divider
  - It integrates a high-performance integer multiplier and divider.
- Floating Processing Unit (FPU)
  - It integrates a double-precision floating-point unit (FPU) that supports FP64.
- Supported Privileged Modes
  - Support Machine Mode, Supervisor Mode and User Mode.
- Bus Interfaces
  - Support 64-bit wide standard AXI system bus interface with fixed Clock-Ratio feature for accessing system instruction and data.
- Low-Power Management
  - Support WFI (Wait For Interrupt) and WFE (Wait For Event) scheme to enter sleep mode.
  - Support two-level sleep modes: shallow sleep mode, and deep sleep mode.
- Core-Private Timer Unit (TIMER)
  - 64-bits wide real-time counter.
  - Support the generation of the timer interrupt defined by the RISC-V standard.
  - Support the generation of the precise periodic timer interrupt (can be used as System Tick) with auto clear-to-zero mode.
  - Support the generation of software interrupt defined by the RISC-V standard.
- Enhanced Core Level Interrupt Controller (ECLIC)
  - Support the RISC-V architecturally defined software, timer and external interrupts.
  - Support fixed number of external interrupt sources.

  - Support fixed number of interrupt levels and priorities, and support software dynamically programmable division of interrupt levels and priorities values.
  - Support interrupts preemption based on interrupt levels.
  - Support vectored interrupt processing mode for extremely fast interrupt response .
  - Support fast interrupts tail-chaining mechanism.
- Support PLIC (Platform Level Interrupt Controller)
- Support NMI (Non-Maskable Interrupt)
- Memory Protection
- Memory Management Unit (MMU)
  - Support SV39.
  - 2-level TLB: Main-TLB and Micro-TLB (i-tlb, d-tlb)
- Support Instructions Extended by User
  - Support Nuclei Instruction Co-unit Extension (NICE) scheme to support user to extend custom instructions according to their applications.
- Support Instruction Cache (I-Cache)
  - The Cache size is 32KBytes.
  - 2-way associative structure.
  - Cache Line Size is 64 Bytes.
  - Support to LOCK, INVAL cachelines.
- Support Data Cache (D-Cache)
  - The Cache size is 32KBytes.
  - 2-way associative structure.
  - Cache Line Size is 64 Bytes.
  - Support to LOCK, FLUSH and/or INVAL cachelines.
- Debugging System
  - Support standard 4-wire JTAG and 2-wire cJTAG interface.
  - Support standard RISC-V debug specification (v0.13).
  - Support mature interactive debugging software/hardware tools, such as GDB, OpenOCD, Lauterbach TRACE32, Segger J-Link, IAR, etc.
- Software Development Tools
  - The Nuclei Hummingbird E603 support the RISC-V standard compilation toolchain and Integrated Development Environment (IDE) on Linux/Windows systems.

4.2 Supported Instruction Set and Privileged Architecture

The Nuclei Hummingbird E603 is following the Nuclei RISC-V Instruction Set and Privileged Architecture Specification <Nuclei_E603_ISA_Spec>, user can easily get the specification from "Nuclei User Center" website http://user.nucleisys.com.

4.3 Top Diagram

The top diagram of Nuclei Hummingbird E603 is as depicted in *The top diagram of Nuclei Hummingbird E603 Core* (page 8), the key points of which are:

- Core Wrapper is the top module of the Core, the sub-modules are:
  - Core: The core part.
  - Reset Sync: The module to sync external async reset signal to synced reset with "Asynchronously assert and synchronously de-assert" style.
  - DEBUG: The module to handle the debug functionalities interactive with JTAG.
- uCore is under Core hierarchy, it is the main part of Core.
- Besides the uCore, there are several other sub-modules:
  - ECLIC: The private interrupt controller.
  - TIMER: The private timer unit.
  - BIU: The bus interface unit.
  - Misc Ctrl: Other miscellaneous modules.

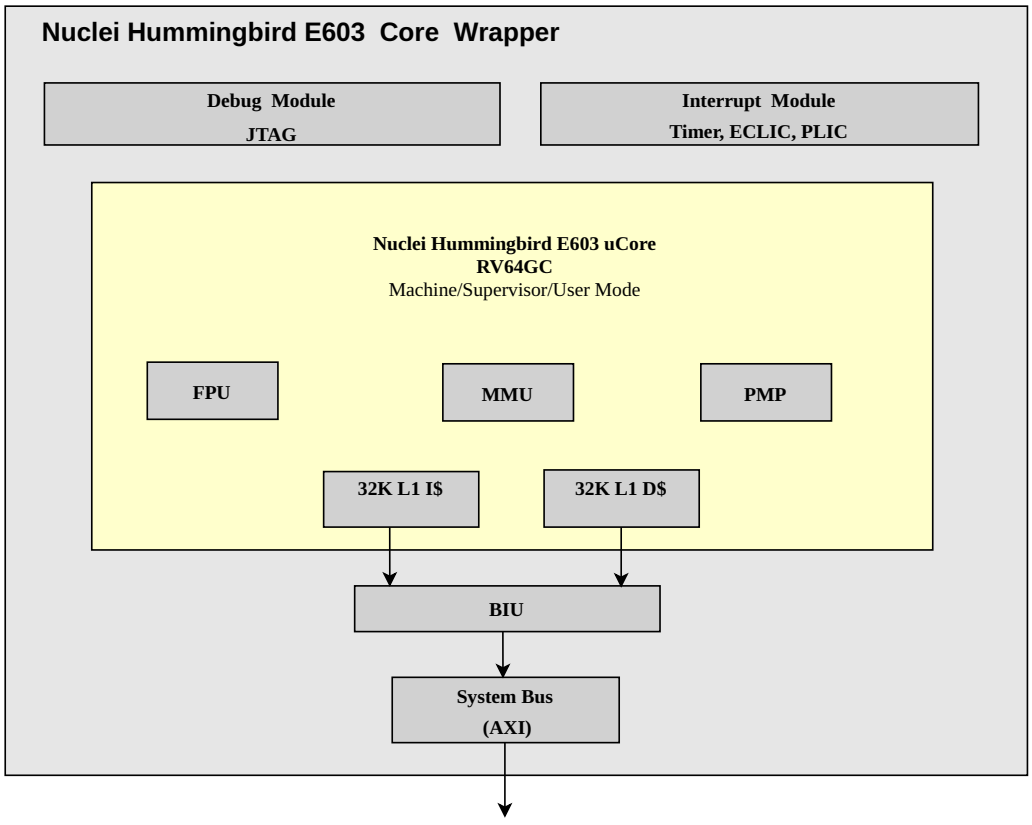

Fig. 4.1: The top diagram of Nuclei Hummingbird E603 Core

5 Functional Introductions

5.1 Clock Domains

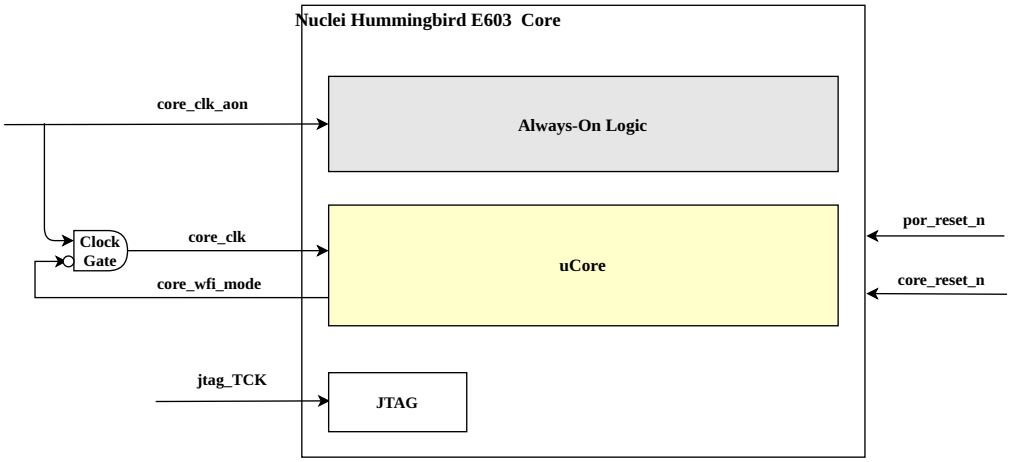

Fig. 5.1: Clock domains of the Nuclei Hummingbird E603 Core

The clock domains of the Nuclei Hummingbird E603 Core are shown in *Clock domains of the Nuclei Hummingbird E603 Core* (page 9). The entire Core is divided into two asynchronous clock domains:

- The main clock domain (for the core_clk and core_clk_aon), which drive most of the functional logics of the Core. Note:
  - core_clk and core_clk_aon have the same frequency and same phases as they are supposed to be clocks from the same clock source.
  - core_clk is the main working clock that drives the main domain inside the Core, core_clk is supposed to be clock-gated at the SoC level during the sleep mode.
  - core_clk_aon is an always-on clock that drives the always-on logic in the Core, including the ECLIC, TIMER, DEBUG, etc.
- The JTAG_CLK clock domain (for the jtag_TCK), which drives the JTAG logics of DEBUG unit.
  - For generated clock tck_s inside JTAG, which is a divide-by-3 clock of jtag_TCK, should be also created in synthesis. More details could be referred to in <Nuclei_Processor_Integration_Guide> section 7.2.

The above two clock domains are asynchronous with each other, so asynchronous cross-clock domain processing has been implemented in the Core.

5.2 Power Domains

There are no power domains specified inside Nuclei Hummingbird E603. Per different applications, the SoC integrator can choose to divide the power domains according to the convenience of the hierarchies inside the Core.

5.3 Core Interfaces

- Clock and Reset
- JTAG/cJTAG
- Interrupt
- Bus
- Misc.

Please refer to *Interfaces for E603* (page 15) for the details of interfaces.

5.4 Memory Resources

The Nuclei Hummingbird E603 supports the following memory resources:

- I-Cache:
  - The Core supports I-Cache (Instruction Cache) to cache the instruction fetched from MEM interface.
  - I-Cache is 2-ways associative structure, with Line Size as 64 Bytes. The size of I-Cache is fixed.
  - Some extended CSR registers are defined for I-Cache Control and Maintenance (CCM) operations, which can be used to INVAL I-Cache by ADDR or by ALL. Please refer to the <Nuclei_CCM_Mechanism> for more details, user can easily get the specification from "Nuclei User Center" website http://user.nucleisys.com.
- D-Cache:
  - The Core supports D-Cache (Data Cache) to cache the data fetched from MEM interface with cacheable attribute address space.
  - D-Cache uses Write-Back and Write-Allocate mechanism, which is 2-ways associative structure, with Line Size as 64 Bytes.
  - Some extended CSR registers are defined for D-Cache Control and Maintenance (CCM) operations, which can be used to FLUSH, INVAL, D-Cache by ADDR or by ALL. Please refer to <Nuclei_CCM_Mechanism> for more details, user can easily get the specification from "Nuclei User Center" website http://user.nucleisys.com.

5.5 Private Peripherals

As shown in *The top diagram of Nuclei Hummingbird E603 Core* (page 8), under the Core hierarchy, in addition to the uCore, the following private peripherals are included:

- DEBUG: handle the JTAG interface and related debugging features.
- ECLIC: the Enhanced Core-Level Interrupt Controller.
- TIMER: the private TIMER unit.

The above peripherals are private to the Core and are accessed using memory mapped address.

See *Address Spaces of Interfaces and Private Peripherals* (page 11) for the details of their specific address space allocation.

5.6 Address Spaces of Interfaces and Private Peripherals

There are quite several interfaces and private peripherals for the Core, the address spaces of them are shown in *Address_Spaces_of_the_Core* (page 11).

Table 5.1: Address_Spaces_of_the_Core

Unit	Base Address	Size	Attribute	Description
DEBUG	Fixed	0x000~ 0xFFF	Device	Address space of DEBUG unit.
ECLIC	Fixed	0x0000 ~ 0xFFFF	Device	Address space of ECLIC unit.
PLIC	Fixed	0x0 ~ 0x3FFFFFF	Device	Address space of PLIC unit.
TIMER	Fixed	0x000 ~ 0xFFF	Device	Address space of TIMER unit.
MEM	N/A	N/A	N/A	The address space other than the above mentioned spaces are all to MEM (System Memory) interface. MEM address space can be: 1. Device, configurable; 2. Non-Cacheable, configurable; 3. Cacheable, configurable;-

5.7 Debug Support

The Nuclei Hummingbird E603 supports standard 4-wire JTAG and 2-wire cJTAG interface, standard RISC-V debug specification (v0.13), fixed number of Hardware Breakpoints, and mature interactive debugging software/hardware tools, such as GDB, OpenOCD, Segger J-Link, IAR, etc.

The Nuclei Hummingbird E603 defines an i signal i_dbg_stop, which can be controlled by the SoC level to disable the debug functionality or not:

- If the value of the i_dbg_stop signal is 0, the debug functionality of the Core is working properly.
- If the value of the i_dbg_stop signal is 1, the debug functionality of the Core is off, and the external Debug Hardware Probe cannot debug the Core through JTAG interface anymore.

5.8 Interrupt and Exception Mechanism

For a detailed description of the Core's interrupt and exception mechanisms, please refer to Nuclei ISA specification <Nuclei_E603_ISA_Spec>, user can easily get the specification from "Nuclei User Center" website http://user.nucleisys.com.

5.9 NMI Mechanism

NMI (Non-Maskable Interrupt) is a special input signal of the Core, often used to indicate system-level emergency errors (such as external hardware failures, etc.). After encountering the NMI, the Core should abort execution of the current program immediately and process the NMI handler instead. For a detailed description of the NMI mechanism of the Nuclei Hummingbird E603 Core, please refer to Nuclei ISA specification <Nuclei_E603_ISA_Spec>, user can easily get the specification from "Nuclei User Center" website http://user.nucleisys.com.

5.10 Control and Status Registers (CSRs)

Some control and status registers (CSRs) are defined in the RISC-V architecture to configure or record the status of execution. CSR registers are registers internal to the Core that uses their private 12-bit encoding space to access.

Some extended CSR registers are defined for I/D-Cache Control and Maintenance (CCM) operations, which can be used to FLUSH, INVAL, LOCK and UNLOCK I/D-Cache by ADDR or by ALL.

For a detailed description of the Core's CSRs, please refer to Nuclei ISA specification <Nuclei_E603_ISA_Spec>, user can easily get the specification from "Nuclei User Center" website http://user.nucleisys.com.

5.11 Performance Monitor Unit

The Nuclei Hummingbird E603 implement full performance monitor methods defined by RISC-V Spec as followed:

- Cycle Counter:
  - A 64-bit wide clock cycle counter that reflects how many clock cycles the Core has executed. This counter will continuously increment as long as the Core's core_clk_aon is ON.
  - The CSR mcycle reflect the lower 32 bits of the counter, and the CSR mcycleh reflect the upper 32 bits of the counter. Please refer to Nuclei ISA specification <Nuclei_E603_ISA_Spec> for the details; user can easily get the specification from "Nuclei User Center" website http://user.nucleisys.com.
- Instruction Retirement Counter:
  - The RISC-V architecture defines a 64-bit wide instruction completion counter that reflects how many instructions the Core successfully executed. This counter will increment if the processor executes an instruction successfully.
  - The CSR minstret reflect the lower 32 bits of the counter, and the CSR minstreth reflect the upper 32 bits of the counter. Please refer to Nuclei ISA specification <Nuclei_E603_ISA_Spec> for more details; user can easily get the specification from "Nuclei User Center" website http://user.nucleisys.com.
- Other User Selected Event Counters
  - Please refer to Section 18 of Nuclei ISA specification <Nuclei_E603_ISA_Spec> for more details.

The Cycle Counter and the Instruction Retirement Counter are typically used to measure performance.

By default, the counter is zero value after a reset and then increments itself continuously. But in Nuclei Nuclei Hummingbird E603 Core, considering the counter increases the power consumption, there is an extra bit in the customized CSR mcountinhibit that can pause the counter to save power when users don't need to monitor the performance through the counter.

Please refer to Nuclei ISA specification <Nuclei_E603_ISA_Spec> for more details; user can easily get the specification from "Nuclei User Center" website http://user.nucleisys.com.

5.12 TIMER Unit

The RISC-V architecture defines a 64-bit Timer Counter which is clocked by the system's low-speed Real Time Clock frequency. The value of this timer is reflected in the register mtime. The RISC-V architecture also defines a 64-bit mtimecmp register that used as a comparison value for the timer. A timer interrupt is generated if the value of mtime is greater than or equal to the value of mtimecmp.

Note: The RISC-V architecture does not define the mtime and mtimecmp registers as CSR registers, but rather as Memory Address Mapped system registers. The specific memory mapped address is not defined by the RISC-V architecture, so it is defined by the implementation. In the 600 Series Core, mtime and mtimecmp are both implemented in the TIMER Unit.

Besides, the TIMER Unit of Nuclei Hummingbird E603 Core can also generate the periodic timer interrupt (normally as System Tick) and the software interrupt.

By default, the counter is zero value after a reset and then increments itself continuously. But in Nuclei Nuclei Hummingbird E603 Core, considering the counter increases the power consumption, there is an extra bit in the customized CSR

mcountinhibit that can pause the counter to save power when users don't need to monitor the performance through the counter.

In debug mode, timer will stop counting when executing codes inserted through debugger to reflect the real behaviors of the program under debugging.

Please refer to Nuclei ISA specification <Nuclei_E603_ISA_Spec> for more details; user can easily get the specification from "Nuclei User Center" website http://user.nucleisys.com.

5.13 Low-Power Mechanism

The low-power mechanism of the Nuclei Hummingbird E603 Core is as below:

- The clocks of the main units in the Core are automatically gated off when they are in idle state to reduce static power consumption.
- The Core supports different sleep modes (shallow sleep mode or deep sleep mode) through WFI (Wait for Interrupt) and WFE (Wait for Event) mechanisms to achieve lower dynamic and static power consumption. For more details about "Wait for Interrupt" and "Wait for Event" mechanism, please refer to Nuclei ISA specification <Nuclei_E603_ISA_Spec> for more details; user can easily get the specification from "Nuclei User Center" website http://user.nucleisys.com.

5.13.1 Clock Control of Entering Sleep Modes

The key points of the clock control (reference scheme) of the core entering sleep mode are as the followings:

- As shown in *Clock domains of the Nuclei Hummingbird E603 Core* (page 9), when the WFI/WFE is successfully executed, the ou signal core_wfi_mode of the Core is asserted, indicating that the Core has entered to the sleep mode; at the SoC level, the signal core_wfi_mode should be used to control the external gate logic to disable the core_clk.
- If the Core enters the deep sleep mode (core_sleep_value is 1), then SoC can decide whether to disable the always on clock core_clk_aon according to its actual scenario.

5.13.2 Clock Control of Exiting Sleep Mode

The Core can be waked up by interrupt, event, debug or NMI. Please refer to Nuclei ISA specification <Nuclei_E603_ISA_Spec> for more details; user can easily get the specification from "Nuclei User Center" website http://user.nucleisys.com.

The key points of the clock control of the Core exiting the sleep mode are as the followings:

- The output signal core_wfi_mode will be de-asserted immediately after the core being waked up. Assuming the SoC control the gate of core_clk using the signal core_wfi_mode, the working clock of core, core_clk will be enabled as soon as the signal core_wfi_mode is de-asserted.
- For the case that the core is waiting for an interrupt to wake up, because the Core can only handle the interrupt processed and distributed by ECLIC unit, then only the interrupt, which is enabled and has greater interrupt level than the interrupt threshold level, can wake up the core. Furthermore, whether enable the core_clk_aon inside the core needs to be handled carefully:
  - As mentioned in *Clock Domains* (page 9), the TIMER unit is clocked by core_clk_aon, so if the SoC system has disabled the always-on clock core_clk_aon, the TIMER unit cannot generate timer or software interrupt because it has no working clock, and the core cannot be woken up.
  - As mentioned in *Clock Domains* (page 9), the ECLIC unit is clocked by core_clk_aon, so if the SoC system has disabled the always-on clock core_clk_aon, then the external interrupt signal must keep asserted until the SoC enable the core_clk_aon again. Otherwise, the ECLIC unit cannot sample the external interrupt signal because it has no working clock, and the core cannot be woken up.
- For the case that the core is waiting for an event or NMI to wake up, if the core sampled (by the core_clk_aon) the i signal rx_evt (indicate there is one event) or the i signal nmi (indicate there is one NMI), the core will be woken up. Furthermore, whether enable the core_clk_aon inside the core needs to be handled carefully:

- If the SoC system has disabled the always-on clock core_clk_aon, then the i signal rx_evt or nmi must keep as 1 until the SoC turns on the clock core_clk_aon. Otherwise, the core cannot sample the rx_evt or nmi as the sample logic has no working clock, and the core will not wake up.

5.14 Physical Memory Protection

In order to perform memory access protection and isolation according to memory physical address and execution privilege mode, the RISC-V standard architecture defines a physical memory protection mechanism: Physical Memory Protection (PMP).

The Nuclei Hummingbird E603 Core is fixed to support the PMP feature. About the programming mode of PMP, please refer to Nuclei ISA specification <Nuclei_E603_ISA_Spec>, user can easily get the specification from "Nuclei User Center" website http://user.nucleisys.com.

5.15 Memory Management Unit (MMU)

The core can have MMU with Linux capable applications, which implements the SV39 mode defined in RISC-V privileged specification, to support Page-Based 39-bit Virtual-Memory System, which can be used for converting Virtual-Address to Physical-Address and corresponding Permission Checking.

MMU has two level TLB (Translation Lookaside Buffer) implemented to cache the page tables for fast subsequent accessing: Main-TLB and Micro-TLB (i-tlb, d-tlb); Main-TLB is the shared TLB for both instruction and data page table, Main-TLB is fixed as 64 entries; Micro-TLB (i/d-tlb) is dedicate for instruction/data page table, each has 8 entries; Micro-TLB (i/d-tlb) will be accessed first, if miss then Main-TLB will be accessed.

MMU supports 4KB, 2MB and 1GB page types, which uses Hardware Translation Table Walk mechanism to fetch page tables from memory when Main-TLB miss without software handling.

Please refer to Nuclei ISA specification <Nuclei_E603_ISA_Spec> for more details, user can easily get the specification from "Nuclei User Center" website http://user.nucleisys.com.

5.16 FPU Feature

The Nuclei Hummingbird E603 Core supports double-precision (D extension) floating point instructions. For the details of D extension, please refer to Nuclei RISC-V Instruction Set and Privileged Architecture Specification <Nuclei_E603_ISA_Spec>, user can easily get the specification from "Nuclei User Center" website http://user.nucleisys.com.

> The Nuclei Hummingbird E603 supports user to add their custom instructions with NICE (Nuclei Instruction Co-unit Extension) interface. Please refer to another document <Nuclei_NICE_Extension> for the details. User can easily get the copy from "Nuclei User Center" website http://user.nucleisys.com.

5.17 NICE Feature

The E603 Core can support user to add their custom instructions with NICE (Nuclei Instruction Co-unit Extension) interface. Please refer to another document <Nuclei_E603_ISA_Spec> for the details.

6 Interfaces for E603

If the E603 is configured as "Single Core Configuration", the interfaces are as below.

6.1 Clock and Reset Interface

The clock and reset signals of E603 Core are as depicted *Clock and Reset Signals for Single Core Configuration* (page 15)

Table 6.1: Clock and Reset Signals for Single Core Configuration

Signal Name	Direction	Width	Description
core_clk_aon	Input	1	This clock is to drive the all logics of the Core, please refer to *Clock Domains* (page 9) for more details.
por_reset_n	Input	1	Power on Reset. This signal is active low signal. This reset will reset the entire Core including the JTAG logics. Note: this reset signal will be synced inside the core to make it as "asynchronously asserted and synchronously de-asserted" style.
core_reset_n	Input	1	System Reset. This signal is active low signal. This reset will reset the Core except the JTAG logics. Note: this reset signal will be synced inside E603 Core to make it as "asynchronously asserted and synchronously de-asserted" style.
reset_bypass	Input	1	If the reset_bypass signal is high, then the internal generated reset will be bypassed, to allow DFT (Design For Test) rules. Note: if the reset_bypass is high, the core_reset_n will be bypassed and disabled, only the por_reset_n will really take effects.
clkgate_bypass	Input	1	If the clkgate_bypass is high, the clock gater will be bypassed, to allow DFT (Design For Test) rules.

6.2 Debug Interface

For a detailed description of Debug interface, please refer to <Nuclei_CPU_Debug_Function_Specification.pdf>.

6.3 Interrupt Interface

The interrupt signals are as depicted *Interrupt and NMI signals for Single Core Configuration* (page 16).

Table 6.2: Interrupt and NMI signals for Single Core Configuration

Signal Name	Direction	Width	Description
nmi	Input	1	NMI (Non-Maskable Interrupt) input. Note: • nmi_irq signal will not be synced inside the Core, so this signal need to be synced at the SoC level if it is from different clock domain at SoC. • Please refer to *NMI Mechanism* (page 11) for the details of NMI.
irq_i	Input	CFG_IRQ_NUM	External Interrupt input, each bit of which can be used to connect an interrupt source. Note: • irq_i signal will not be synced inside the Core, so these signals need to be synced at the SoC level if it is from different clock domain at SoC.

6.4 Bus Interfaces

The E603 Core Configuration supports several bus interfaces, including:

- MEM (System Memory) Interface.

6.4.1 MEM Interface

MEM interface is used to access system memory for instruction and data. MEM interface is AXI protocol interface as shown in below tables.

Table 6.3: MEM AXI signals for Single Core Configuration

Signal Name	Direction	Bit width	Description
mem_clk_en	Input	1	MEM interface clock ratio
mem_arvalid	Output	1	AXI protocol ARVALID signal
mem_arid	Output	3/4	AXI protocol ARID signal.
mem_araddr	Output	PA_SIZE	AXI protocol ARADDR signal
mem_arlen	Output	8	AXI protocol ARLEN signal (0-255)
mem_arsize	Output	3	AXI protocol ARSIZE signal (0-7)
mem_arburst	Output	2	AXI protocol ARBURST signal (0-2)
mem_arlock	Output	1	AXI protocol ARLOCK signal
mem_arcache	Output	4	AXI protocol ARCACHE signal
mem_arprot	Output	3	AXI protocol ARPROT signal
mem_arready	Input	1	AXI protocol ARREADY signal
mem_arsmode	Output	1	Nuclei sideband signal, s-mode indicator in AR Channel
mem_ardmode	Output	1	Nuclei sideband signal, d-mode indicator in AR Channel
mem_awvalid	Output	1	AXI protocol AWVALID signal
mem_awid	Output	3/4	AXI protocol AWID signal.
mem_awaddr	Output	PA_SIZE	AXI protocol AWADDR signal
mem_awlen	Output	8	AXI protocol AWLEN signal (0-255)
mem_awsize	Output	3	AXI protocol AWSIZE signal (0-7)

continues on next page

Table 6.3 – continued from previous page

Signal Name	Direction	Bit width	Description
mem_awburst	Output	2	AXI protocol AWBURST signal (0-2)
mem_awlock	Output	1	AXI protocol AWLOCK signal
mem_awcache	Output	4	AXI protocol AWCACHE signal
mem_awprot	Output	3	AXI protocol AWPROT signal
mem_awready	Input	1	AXI protocol AWREADY signal
mem_awsmode	Output	1	Nuclei sideband signal, s-mode indicator in AW Channel
mem_awdmode	Output	1	Nuclei sideband signal, d-mode indicator in AW Channel
mem_wvalid	Output	1	AXI protocol WVALID signal
mem_wdata	Output	MEM_AXI_DATA_WIDTH	AXI protocol WDATA signal
mem_wlast	Output	1	AXI protocol WLAST signal
mem_wstrb	Output	16	AXI protocol WSTRB signal
mem_wready	Input	1	AXI protocol WREADY signal
mem_rvalid	Input	1	AXI protocol RVALID signal
mem_rid	Input	3/4	AXI protocol RID signal.
mem_rdata	Input	MEM_AXI_DATA_WIDTH	AXI protocol RDATA signal
mem_rlast	Input	1	AXI protocol RLAST signal
mem_rresp	Input	2	AXI protocol RRESP signal
mem_rready	Output	1	AXI protocol RREADY signal
mem_bvalid	Input	1	AXI protocol BVALID signal
mem_bid	Input	3/4	AXI protocol BID signal.
mem_bresp	Input	2	AXI protocol BRESP signal
mem_bready	Output	1	AXI protocol BREADY signal

The data width of the MEM interface is fixed at 64 bits

Note:

- For AR/AW channel:
  - D-Cache Writeback/Eviction will trigger INCR burst type;
  - D-Cache Read/Write Miss will trigger WRAP burst type;
  - I-Cache Read Miss will trigger WRAP burst type;
  - Maximum non-cacheable/device read outstanding capacity is 8;
  - Maximum non-cacheable/device write outstanding capacity is 8 (If write-merge is enable, it is more than 8);
  - Maximum cacheable outstanding is 2 (If hardware prefetch is enable, it is 3);
  - Maximum icache linefill oustanding is 2;
  - AxPROT[0] can be 1 (machine mode) or 0 (other privilege mode); AxPROT[2] can be 1 (instruciton access) or 0 (data access).
  - AxLock can be 1 (exclusive access) when core use Atomic Instruction to access an address of non-cacheable/device PMA.
  - Axcache is 4b'0000 for Device Region and 4b'0011 for NoC Region and 4b'1111 for Cacheable Region.
- Core will default keep order between Device Read and Write, and E603 core can be enable device store non-blocking, and maximum device store outstanding is 8 too.
- AXI ID is 4 when L1 DCache supports prefetch feature or it is 3. The AXI ID of NoC & Device Region is the same.

- About the xxx_clk_en signal, it generally uses Clock Gating enable based integer divider, when the xxx_clk_en is 1 constantly, then the genrated clock is same with source clock; if the xxx_clk_en is high every 1 orignal clock period, then it is divide by 2 clock, if the xxx_clk_en is high every 2 orignal clock period, it is divide by 3 clock, and etc.
- Follows shows a example waveform of Divide by 2 clock generation:

- Follows shows a example waveform of Divide by 3 clock generation:

Note: Nuclei Core is OK for CG based clock divider or other clock divider, user just needs to keep the phase relationship as the waveform shows.

6.5 NICE Interface

The NICE interface is used to allow user to extend the custom instructions according to their applications. Please refer to another document <Nuclei_E603_ISA_Spec> for the details.

6.6 Other Functional Interface

Table 6.4: Other Interface signals

Signal Name	Direction	width	Description
tx_evt	Output	1	600 Series Core use this txevt output a pulse signal as the transmitting Event signal. Please refer to "Nuclei_RISCV_ISA_Spec" for more details of CSR register txevt.
rx_evt	Input	1	The receiving Event as the event to wake up Core from WFE mode. Please refer to *Low-Power Mechanism* (page 13) for more details of WFE.
mtime_toggle_a	Input	1	The mtime_toggle_a is a periodic pulse signal (normally as System Tick) from the SoC system, and used to drive the counter of the internal TIMER unit inside the Core. Note: • This signal is treated as an asynchronous input signal, and is synchronized within the Core (using several DFF synchronizers). • After the synchronization, both the rising edge and falling edge of the signal are sampled by the core_clk_aon of the Core, and any detected edge will trigger the TIMER counter to increment. • It is recommended that use the output of the register driven by the "slow clock" (e.g., rtc_clk, whose frequency is the divided frequency of core_clk_aon) as the input of this signal. Then the self-increment frequency is equal to the frequency of the "slow clock", as shown in the *mtime_toggle_a signal generation_1* (page 20). Hence, the lower the frequency of the slow clock, the lower the self-increment frequency of the internal timer, the lower the dynamic power consumption.

continues on next page

Table 6.4 – continued from previous page

Signal Name	Direction	width	Description
dbg_toggle_a	Input	1	The dbg_toggle_a is a periodic pulse signal from the SoC system, and used to drive the time-out counter of the DEBUG unit inside the Core. The time-out counter of DEBUG unit is used to protect the case that if the JTAG Debugger Probe is unexpectedly pulled out which leave the DEBUG unit in an uncertain state. Note: • This signal is treated as an asynchronous input signal, and is synchronized within the Core (using several DFF synchronizers). In order to make the time-out upper limit to around 170-320ms, it is recommended to use 25kHz~50kHz system real-time-clock as the slow clock to generate this dbg_toggle_a signal, the generation scheme of which is similar to *mtime_toggle_a signal generation_1* (page 20).
hart_id	Input	10	The Core's HART ID indication signal, when integration in SoC, this input can be tied to a specific constant value (range 0 ~ 1023), and the value of it will be reflected in CSR register mhartid inside the Core. In single core case, this signal is better to tied as zero. In multi cores with jtag chain case, this signal of the cores is better to be continuous from zero.
reset_vector	Input	64	This signal is to indicate the PC value of the first instruction to be fetched after reset. User can use this signal at SoC level to control the PC address of first instruction executed after reset.
hart_halted	Output	1	If this output signal is 1, it is indicating the Core is under debug mode.
i_dbg_stop	Input	1	If this input signal is 1, then the Core's Debug functionality will be disabled, and the external Debug Hardware Probe cannot debug the Core through JTAG interface.
sysrstreq	Output	1	This output signal is the System Reset request generated from the JTAG. The SoC integrator can use this signal to trigger the Core's core_reset_n to reset the Core except the JTAG logics. Note: 1. please do not trigger por_reset_n; 2. please do not disable core_clk_aon; 3. please delay it one cycle on SoC;
core_wfi_mode	Output	1	If this signal is 1, then indicating the Core is under sleep mode. Please refer to *Low-Power Mechanism* (page 13) for more details of sleep modes.
core_sleep_value	Output	1	This signal is to indicate the shallow sleep or deep sleep mode. Please refer to *Low-Power Mechanism* (page 13) for more details of sleep modes.
override_dm_sleep	Input	1	This signal can be configured in SoC to enable the clock of debug module when deep sleep mode (core_sleep_value=1), to support debug in low-power.

continues on next page

Table 6.4 – continued from previous page

Signal Name	Direction	width	Description
dbg_stoptime	Output	1	dcsr[9], stoptime in debug control and status register: 0: increase timers as usual; 1: don't increase timer while in debug mode; Can be used to stop the SoC timer if necessary.
stop_on_reset	Input	1	This signal is used to STOP the core after reset, to allow other master such as DMA to preload the ILM/DLM through the slave port, this signal is only effective after reset once, it cannot stop the core during running.

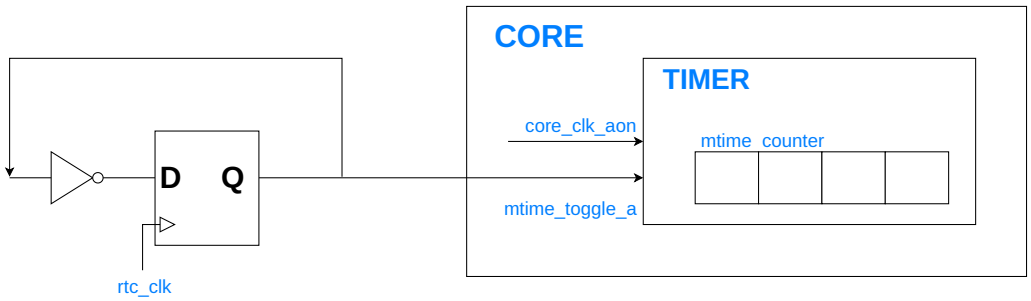

Fig. 6.1: mtime_toggle_a signal generation_1

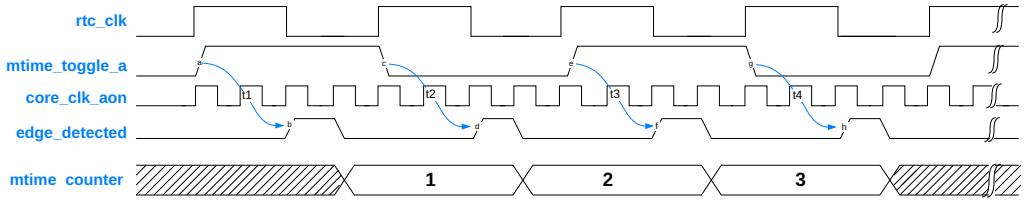

Fig. 6.2: mtime_toggle_a signal generation_2

7 Appendix

- **Nuclei RISC-V IP Products**: https://www.nucleisys.com/product.php
- **Nuclei Spec Documentation**: https://nucleisys.com/download.php#spec
- **Nuclei RISCV Tools and Documents**: https://nucleisys.com/download.php
- **Nuclei Prebuilt Toolchain and IDE**: https://nucleisys.com/download.php#tools
- **NMSIS**: https://github.com/Nuclei-Software/NMSIS
- **Nuclei SDK**: https://github.com/Nuclei-Software/nuclei-sdk
- **Nuclei Linux SDK**: https://github.com/Nuclei-Software/nuclei-linux-sdk
- **Nuclei Software Organization in Github**: https://github.com/Nuclei-Software/
- **RISC-V MCU Organization in Github**: https://github.com/riscv-mcu/
- **RISC-V MCU Community Website**: https://www.rvmcu.com/
- **Nuclei riscv-openocd**: https://github.com/riscv-mcu/riscv-openocd
- **Nuclei riscv-gnu-toolchain**: https://github.com/riscv-mcu/riscv-gnu-toolchain