# GeMRTOS: Multiprocessor RTOS

## User and Development Manual — Chapter 6: GeMRTOS IP for Platform Designer

*gemrtos.com*

August 2026

Revision: 0.9a-2026-06-01

© 2026 GeMRTOS. All rights reserved.

ALTERA, AGILEX, ARRIA, CYCLONE, MAX, STRATIX, MEGACORE, NIOS, QUARTUS and PLATFORM DESIGNER are trademarks of Altera Corporation.

All other words and logos identified as trademarks or service marks are the property of their respective holders.

CHAPTER 6

# GeMRTOS IP for Platform Designer

## 6.1 Introduction

The GeMRTOS Multiprocessor IP component simplifies the design of multiprocessor systems, offering significant advantages over traditional development methods. By reducing the complexity of processor booting, synchronization, and mutual exclusion, the IP component makes designing a multiprocessor system nearly as straightforward as creating a uniprocessor system. This streamlined approach lowers the barrier to entry for developers seeking to harness the power of parallel processing in their embedded designs.

## 6.2 GeMRTOS IP Architecture

The GeMRTOS IP architecture includes all the components required to implement multiprocessor systems along with other components for ease of system design and debugging. As depicted in Figure 6.1, the GeMRTOS Multiprocessor IP incorporates the following components, each designed for a specific purpose:

- **GeMRTOS Controller:** This serves as the core of the GeMRTOS Multiprocessor IP component, responsible for time management and external events. It enables the Avalon-conduit port, providing easy access to external signals.
- **System Processors:** This component supports both Nios II and Nios V soft processors, offering flexibility in processor selection.
- **Nios Monitor Component:** This serves as a debugging tool, sniffing each processor bus to detect special transaction patterns. In configurations without a debugging architecture, this component operates transparently.
- **MM-Avalon Bridge for External Devices:** This bridge facilitates the definition of different clock sources for processors and external devices, enabling efficient clock domain adaptation.
- **JTAG-UART Devices:** These devices, configurable for STDIO and STDERR consoles, streamline system design and debugging.
- **MM-Avalon Bridge for HPS Interface:** This bridge enables access to and from the HPS domain memory map. Knowledge required for use.

Overall, the GeMRTOS Multiprocessor IP component significantly reduces the challenges associated with designing complex multiprocessor systems, thereby streamlining the development process and improving efficiency.

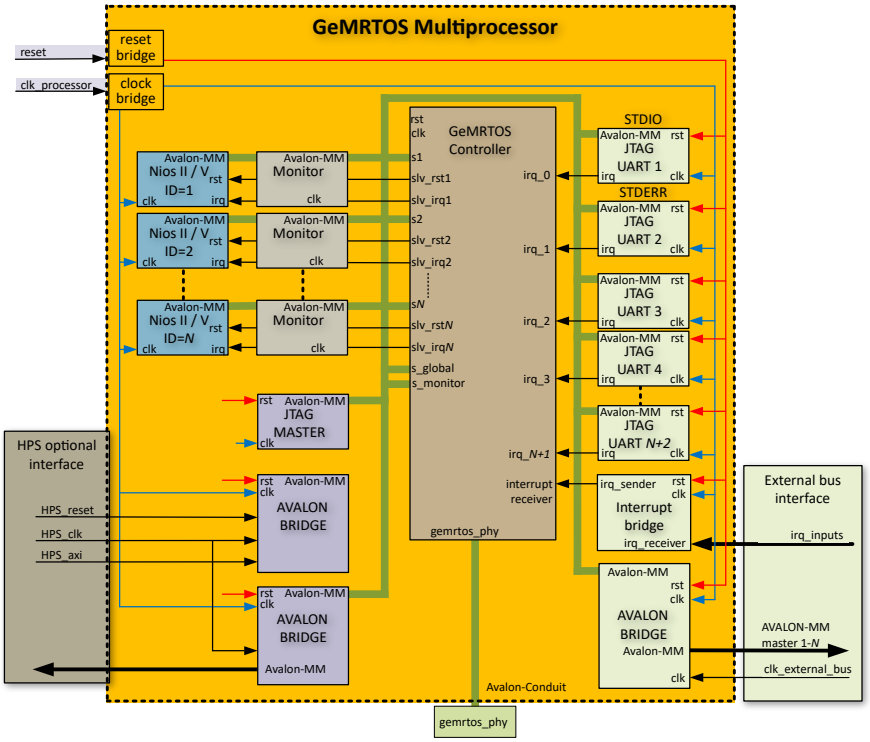


Figure 6.1: GeMRTOS Multiprocessor IP internal architecture.

## 6.3 GeMRTOS IP Configuration Parameters

The GeMRTOS IP component seamlessly integrates with Altera's Platform Designer development tool, appearing as the `GeMRTOS_Multiprocessor` component in the catalog. This integration provides a streamlined and adaptable approach to system configuration. Developers can tailor the GeMRTOS IP component by adjusting the parameters that affect the GeMRTOS Multiprocessor component's operation.

Table 6.1 summarizes the GeMRTOS IP configuration parameters, detailing their ranges and descriptions.

Table 6.1: GeMRTOS IP Configuration Parameters

| Parameter | Range | Description |
|---|---|---|
| **Number of processors** | 1-32 | Determines the number of processors to be included in the GeMRTOS Multiprocessor component. |
| **Processor type** | Nios II/e, Nios II/f, Nios V/m | Determines the type of processor to be utilized. |
| **Reset vector memory** | System memories available | Determines the memory component in which the reset vector of the processor is configured. The parameter may be configured from a list of available options according to the systems memory the component is connected to. |
| **Reset vector offset** | | Determines the offset in the reset vector memory for the reset vector of the processor. |

Continued on next page

Table 6.1: GeMRTOS IP Configuration Parameters (Continued)

| Parameter | Range | Description |
|---|---|---|
| **Exception vector memory (only for Nios II)** | System memories available | Determines the memory component in which the exception vector of the processor is configured. The parameter may be configured from a list of available options according to the systems memory the component is connected to. |
| **Exception vector offset (only for Nios II)** | | Determines the offset in the exception vector memory for the exception vector of the processor. |
| **Instruction cache size** | None, 512Bytes, 1KB, 2KB, 4KB, 8KB, 16KB, 32KB, 64KB. | Set the instruction cache size for each processor of the system when it is supported. |
| **Enable pipeline in CPU (Nios V/m only)** | True / False | Enable the pipeline feature of the CPU when supported. |
| **Independent external processor buses** | True / False | When True, the GeMRTOS IP component produces one bus for each processor, useful to share the bandwidth load over several buses. When False, all the processors share the same bus, useful to simplify the design of the system. It is always advisable to use independent external processor buses to improve the performance of the system. |
| **Include JTAG UART for STDIO** | True / False | When True, a JTAG UART component is included in the GeMRTOS IP configuration for STDIO usage. |
| **Include JTAG UART for STDERR** | True / False | When True, a JTAG UART component is included in the GeMRTOS IP configuration for STDERR usage. |
| **Include JTAG UART for each processor** | True / False | When True, a JTAG UART component is included in the GeMRTOS IP configuration for each processor. This feature may be useful for debugging purposes or for demonstrations. |
| **Enable HPS internal access** | True / False | When True, a bridge is enable for accessing the GeMRTOS IP internal memory map from a HPS. |
| **Quartus Prime project name** | Selection from project folder | Determines the Quartus Prime project name in which the Platform Designer system is instantiated. This configuration is useful to automate the `gemrtos_build` script. |
| **Platform Designer project name** | Selection from project folder | Determines the name of the current Platform Designer project name, useful to automate the `gemrtos_build` script. |

The GeMRTOS Multiprocessor IP parameters (illustrated in Figure 6.2) can be configured through the standard parameter windows of the Platform Designer. These configuration options become accessible when the GeMRTOS component is selected within the System Contents window of the Platform Designer interface.

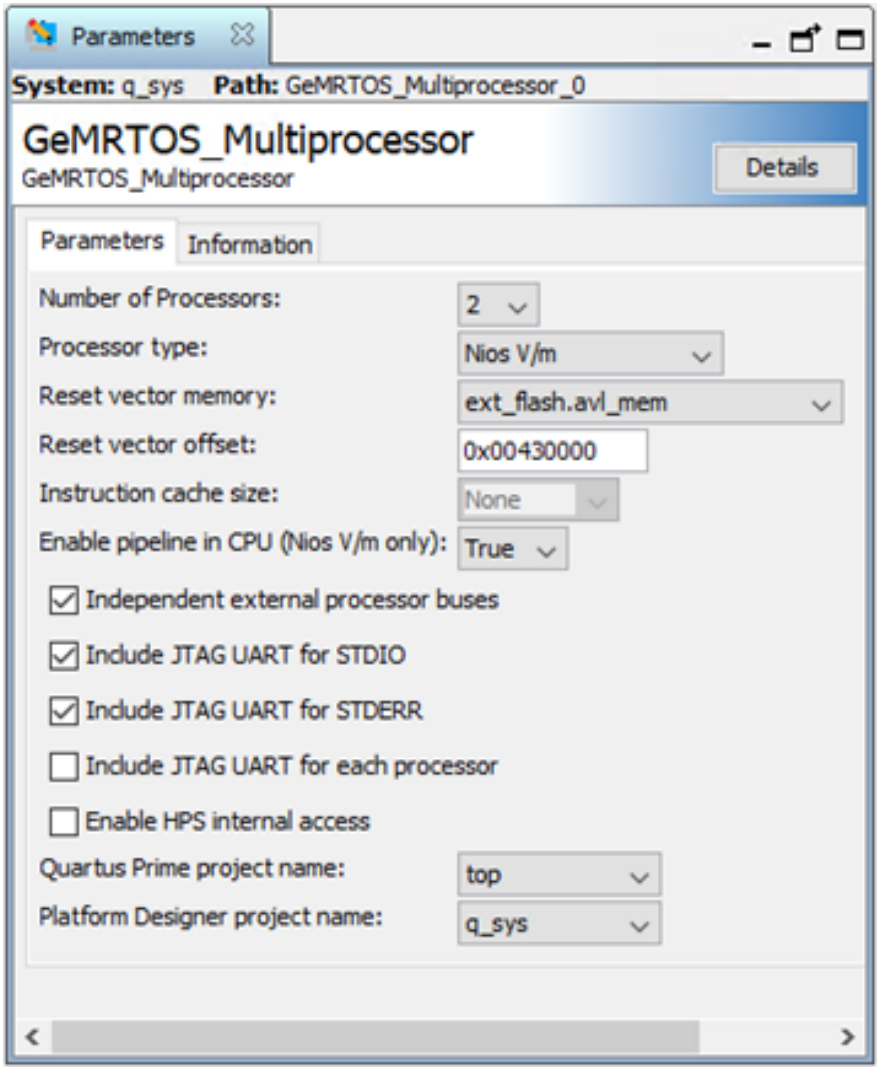


Figure 6.2: GeMRTOS IP parameter configuration tab window.

This setup allows system designers to customize the GeMRTOS Multiprocessor IP to suit their specific project requirements, integrating it effectively within the larger system architecture defined in the Platform Designer tool.

The available parameters and their possible values dynamically adapt based on two key factors:

- Connected Memory Devices: The type, number, and characteristics of memory devices connected to the GeMRTOS component influence which parameters are presented and their available options. For example:
- Different memory types (e.g., SRAM, DRAM, Flash) may expose different configuration options. The size and organization of connected memories could affect addressable ranges or caching parameters.
- Selected Processor Type: The specific processor architecture chosen for the system impacts the available features and, consequently, the relevant configuration parameters.

This adaptive parameter system ensures that:

- Users are presented with only relevant configuration options.
- The component can be optimally configured for the specific hardware environment.
- Incompatible or unavailable options are automatically hidden or disabled.

By dynamically adjusting the parameter set, the GeMRTOS configuration interface streamlines the setup process and helps prevent potential misconfigurations that could arise from presenting irrelevant options.

## 6.4 GeMRTOS IP Information

The GeMRTOS IP Information tab window (Figure 6.3) displays configuration information that is valuable for design and debugging purposes. It provides a quick overview of key system parameters without the need to navigate through multiple configuration windows.

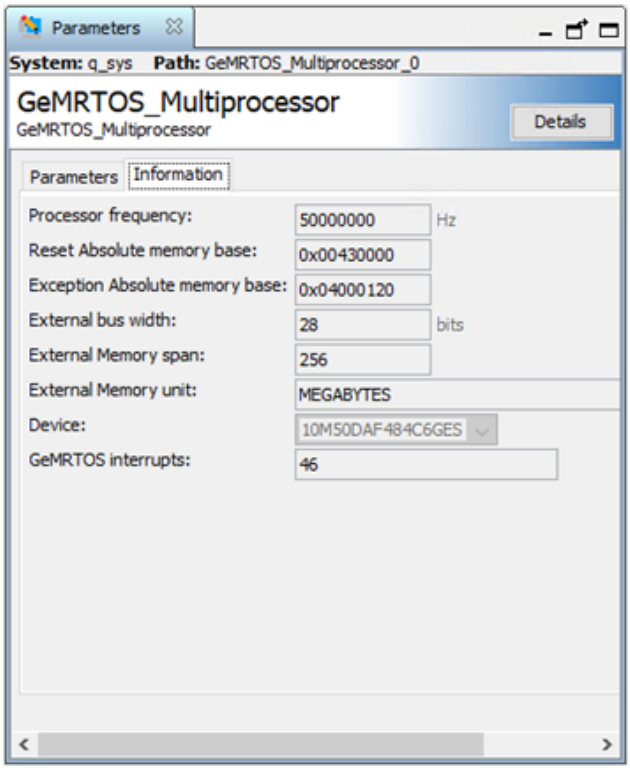


Figure 6.3: GeMRTOS IP Information tab window.

Table 6.2: GeMRTOS IP information parameters

| Parameter | Description |
|---|---|
| `Processor frequency` | Shows the clock frequency connected to the processors. Important for timing calculations and performance estimations. |
| `Reset absolute memory base` | Displays the absolute address where processors reset. Useful when defining addresses in flash memory regions shared with FPGA configuration. |
| `Exception absolute memory base` | Shows the absolute address for processor exceptions. Also helpful for address allocation in shared flash memory regions. |
| `External bus width` | Indicates the width of the bus connecting external components. Automatically updated when components are connected to processor buses. |
| `External Memory span` | Displays the length of the external bus. Measured in units specified in the "External Memory Unit" field. |
| `External Memory unit` | Specifies the unit of measurement for the External Memory span. |
| `Device` | Shows the selected FPGA device for implementation. Crucial for understanding hardware constraints and capabilities. |
| `GEMRTOS Interrupts` | Indicates the number of interrupts configured in the GeMRTOS IP. Important for understanding the system's interrupt handling capacity. |

This information tab serves as a quick reference for system designers and debuggers. It provides a snapshot of critical system parameters, helping to ensure that the configuration aligns with the intended design and assisting in troubleshooting efforts. The data presented here can be particularly useful when defining memory maps, calculating timing requirements, or verifying interrupt configurations.

By presenting this information in an easily accessible format, the GeMRTOS IP information tab streamlines the design process and helps prevent configuration errors that could arise from misunderstanding or overlooking key system parameters.

## 6.5 GeMRTOS IP Memory Map

The GeMRTOS IP handles memory mapping for all its internal components which are connected to the system bus. This section details the memory allocation process and the structure of the internal memory map.

### Memory Map Assignment

- The GeMRTOS IP automatically assigns memory map addresses to all its internal components connected to the system bus.
- The processor with `CPU_ID` 1 determines the system's memory map (useful for BSP project generation).
- The memory addresses assigned by `CPU_ID` 1 are used by the compiler when generating the application.

### Internal Memory Map Components

The internal memory map includes the following components:

Table 6.3: GeMRTOS IP internal memory map.

| Component | Description |
|---|---|
| `processor debug memory` | Memory region dedicated to debug capabilities of the processor, if available. |
| `gemrtos_global_region` | Memory region of the global register of the GeMRTOS controller. |
| `gemrtos_processor_region` | Processor shadow memory region in the GeMRTOS controller. |
| `jtag_uart_0` | JTAG UART component for STDIO when enabled in GeMRTOS parameter window. |
| `jtag_uart_1` | JTAG UART component for STDERR when enabled in GeMRTOS parameter window. |
| `jtag_uart_2..Processors+1` | JTAG UART component for each processor when enabled in GeMRTOS parameter window. |

### 6.5.1 Memory Map Organization

All memory-mapped components listed in Table 6.3 are placed in memory addresses after (higher than) the external components connected to the GeMRTOS IP component in the System Contents window.

This memory mapping scheme ensures:

- Each component has a dedicated address space.
- A consistent memory layout across the system.
- Proper integration of debug and communication features (e.g., JTAG UARTs).
- Correct memory location referencing by the compiler when generating code.

Understanding this memory map is crucial for developers working with GeMRTOS, as it impacts how they access various system components and debug their applications. It also highlights the system's flexibility in accommodating different configurations of processors and communication interfaces.

### 6.5.2 Conclusion

The GeMRTOS IP's memory mapping capabilities are a crucial component of its overall system architecture. By automatically assigning memory addresses to connected components, the GeMRTOS IP ensures a consistent and organized memory layout across the entire system.

The central role of the `CPU_ID` 1 processor in defining the memory map is particularly noteworthy. This design choice allows for a unified memory representation, which is essential for the compiler to correctly generate application code that can seamlessly interact with the various system resources. The memory addresses determined by `CPU_ID` 1 serve as the reference point for the entire software ecosystem, streamlining development and integration efforts.

The internal memory map components, such as the processor debug memory, GeMRTOS global and processor regions, and the JTAG UART interfaces, provide valuable system-level features and capabilities. By strategically placing these components in the memory hierarchy, the GeMRTOS IP ensures that they can be efficiently accessed and utilized by both hardware and software components, enhancing the overall system's functionality

and debugging capabilities.

Ultimately, the GeMRTOS IP's memory mapping approach demonstrates a well-designed and comprehensive strategy for managing the complex interplay of hardware and software within a real-time embedded system. This level of attention to memory organization and accessibility is a testament to the GeMRTOS IP's commitment to delivering a robust and versatile platform for developers working on demanding, time-critical applications.