[laysim-GR712RC-001]

laysim-GR712RC Dual Core Processor Emulator User's Manual

Prepared by : Jong-Wook Choi (jwchoi@kari.re.kr)

Data : 2025-10-23

- Table of Contents -

- Table of Figures -

- Revision History -

Version	Date	Description	Name
Rev #00	2022/05/12	Initial Version with laysim-GR712RC v2.0	Jong-Wook Choi
Rev #01	2025/09/29	Updated according to laysim-GR712RC v3.0	Jong-Wook Choi
Rev #02	2025/10/23	Updated according to laysim-GR712RC v3.1	Jong-Wook Choi

1. Introduction

1.1 General

This document is the user's manual of `laysim-GR712RC` emulator. It describes the architecture and general usage of `laysim-GR712RC`. `laysim-GR712` consists of two types of emulators: `laysim-gr712rc` and `laysim-gr712rc-dbt`. These are provided both as stand-alone applications and as a set of libraries that can be integrated into an existing simulator.

`laysim-gr712rc` is a GUI-based, cycle-accurate emulator for the GR712RC dual core processor using an interpretation method. It includes an embedded source-level debugger and supports the MMU for `XtratuM-SMP`, `AIR` and Linux SMP. `laysim-gr712rc-dbt` is a GUI-based high-performance, cycle-approximate processor emulator that uses dynamic binary translation (DBT). `laysim-GR712RC` supports both Windows and Linux x64 platforms, including CentOS 7/8, Ubuntu 18.04/22.04, openSUSE 12/15 and Debian 10.3.

Also the CLI (Command Line Interface) versions are provided separately. The CLI versions have command line interfaces without a GUI, so the performance of CLI version shows higher than the GUI version of `laysim-GR712RC`.

Figure 1-1 shows a brief comparison of the three versions of `laysim-GR712RC`.

Figure 1-1. Quick Comparison laysim-GR712RC

Core	laysim-gr712rc	laysim-gr712rc-dbt
Emulator Type	Interpreter - For S/W development and debugging	DBT (Dynamic Binary Translation) - Aim for fast execution and simulation
Cycle Accuracy	Cycle accurate Max ±5% error rate	Cycle approximate Max ±10% error rate (But the error rate increases on FPU emulation)
L1 Cache	Fully implemented	No L1 cache for speed-up
Multicore Emulation	Instruction & Cycle level - Execute a instruction from the lowest local cycles of core every time	Round-robin fashion on time quantum (Default time quantum is 200) - 1 means 10nsec on 100MHz, so 200 is 2usec - The time quantum can be changed using '-quantum' start-up option
Control Console	Support - Can control emulation and check status of cores and memory - Can check the performance and information	Support - Can control emulation and check status of cores and memory - Can check the performance and information
GDB	Fully Supported - laysim-gr712rc also includes the embedded source-level debugger, so it is possible to debug GR712RC application without GDB.	Fully Supported
Real-Time Performance (RTP) **i9-12900K @5Gz**	1) Single Core - 113.84% RTP on BCC2 dhrystone - 100.96% RTP on BCC2 whetstone SP 2) Multi Core - 41.02% RTP on smpmigration02 - 40.14% RTP on smpmrsp01	1) Single Core - 1409.50% RTP on BCC2 dhrystone - 357.54% RTP on BCC2 whetstone SP 2) Multi Core - 465.47% RTP on smpmigration02 - 330.42% RTP on smpmrsp01

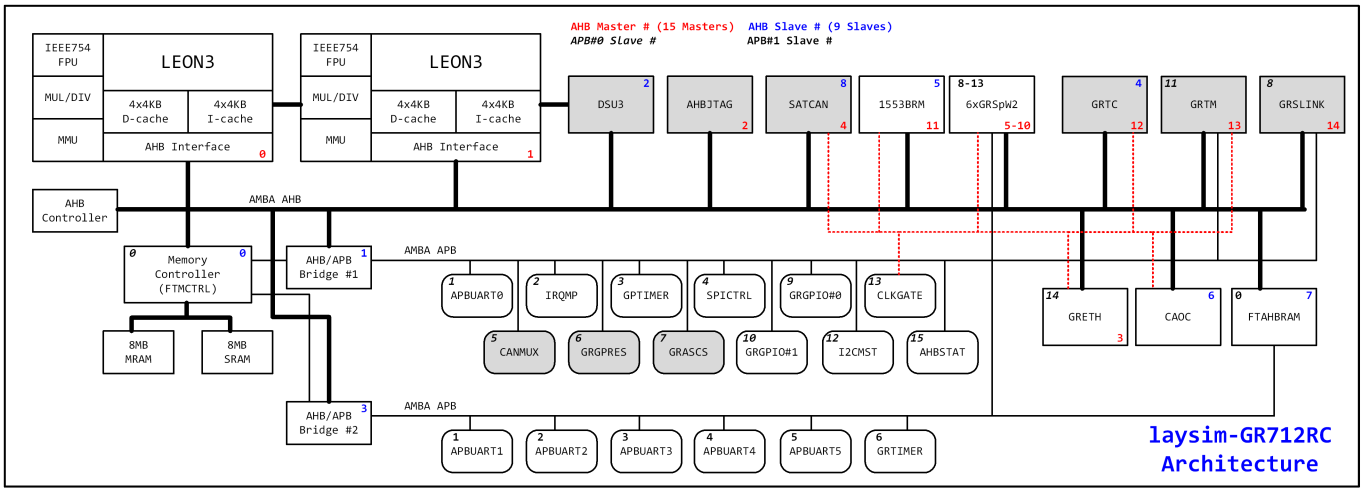
1.2 laysim-GR712RC Overall Architecture

The architecture of `laysim-GR712RC` is shown in Figure 1-2 and it emulates the GR712RC dual core processor with GRLIB devices of GR712RC.

- LEON3 dual core processor (0x01:0x053) with 100MHz clock

- Dual LEON3 with MMU and GRFPU
- 16KB 4-way instruction cache and 16KB 4-way data cache only for `laysim-gr712rc`
- Supported GRLIB devices of GR712RC
  - IRQMP (0x01:0x00D), FTMCTRL (0x01:0x054), AHBSTAT (0x01:0x052), GRSpW2 (0x01:0x029),
  - B1553BRM (0x01:0x072, BC/RT), CANOC (0x01:0x019), GRETH (0x01:0x01D), APBUART (0x01:0x00C)
  - GPTIMER (0x01:0x011), GRTIMER (0x01:0x038), GRGPIO (0x01:0x01A), SPICTRL (0x01:0x02D)
  - I2CMST (0x01:0x28), CLKGATE (0x01:0x02C), FTAHBRAM (0x01:0x050)
- Unsupported GRLIB devices of GR712RC
  - GRTC (0x01:0x031), GRTM (0x01:0x030), CANMUX (0x01:0x081), GRASCS (0x01:0x043)
  - GRGPREG (0x01:0x087) , GRSLINK (0x01:0x02F), SATCAN (0x01:0x080)
- Memory : 8MB MRAM in ROM and 8MB SRAM in RAM as default
- CPU Frequency, ROM size, RAM size, and APBUART#0 baudrate can be changed using start-up option.

Figure 1-2. Architecture of laysim-GR712RC



1.3 Architecture of laysim-gr712rc-dbt

`laysim-gr712rc-dbt` uses only the dynamic binary translator without an interpreter, and it incorporates several performance enhancement techniques, including the reduction of generated machine codes, minimization of helper functions, and translation block chaining. It also adopts the shadow memory to handle the self-modifying code. The architecture of `laysim-gr712rc-dbt` is illustrated in Figure 1-3.

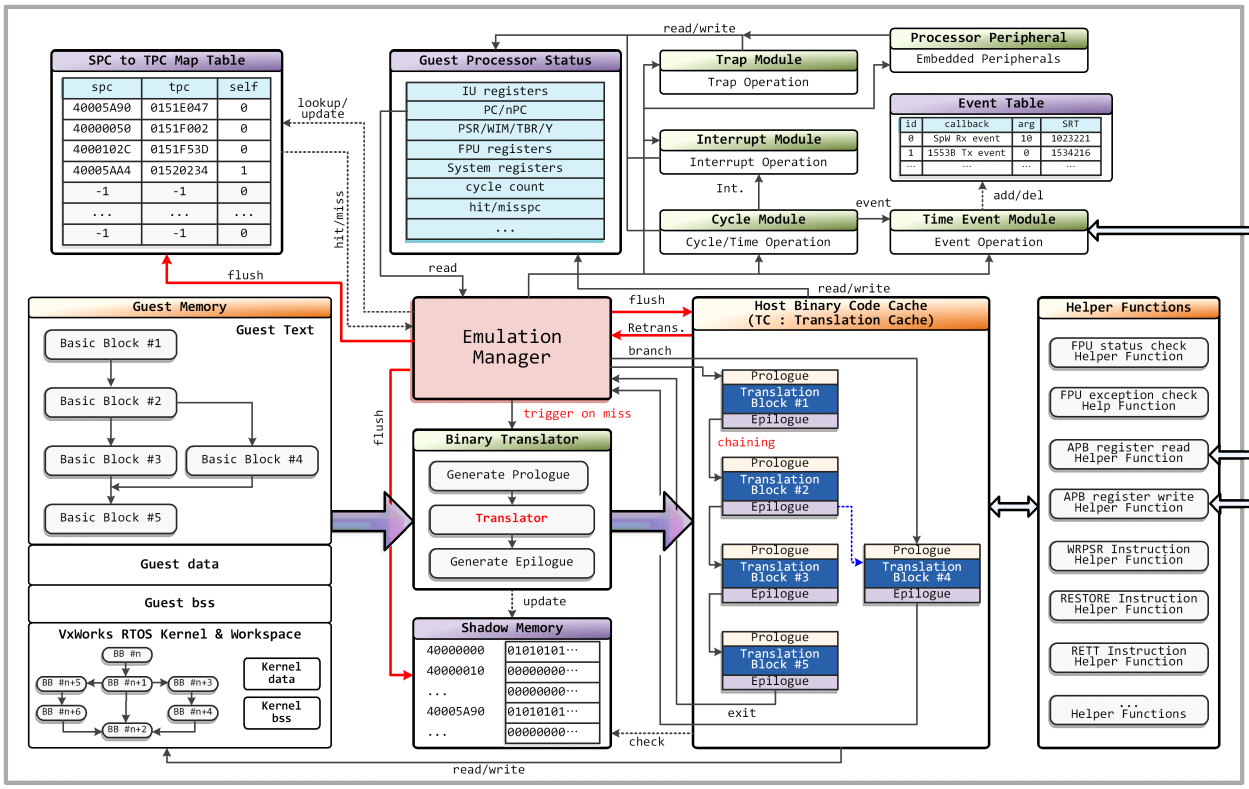
The emulation manager (EM) controls all operations of the `laysim-gr712rc-dbt`. It activates the DBT engine, which is responsible for generating native codes (x64) from guest codes (SPARC) when the TPC (target program counter) is not found in the SCP-to-TPC (source program counter to target program counter) map table. The SPC-to-TCP map table provides an index into the translation cache, which is implemented as a direct mapping table, and the TC (translation cache) holds the translation blocks.

The shadow memory maintains the translation status of guest memory to check for self-modifying conditions. Helper functions, which are called from TB (translation block), support complex guest instructions and enable communication with the the simulator; these functions are

minimized to improve performance.

The cycle module maintains the simulation time and accounts for instruction execution time. It also manages the processor's timer devices and invokes timer interrupts on expiration. The time event module services scheduled events in the event table. The interrupt module checks interrupt conditions and manipulates the interrupt controller of the processor. All synchronous traps and interrupts are handled by the trap module.

Figure 1-3. Architecture of laysim-gr712rc-dbt



1.3.1 Instruction Cycles

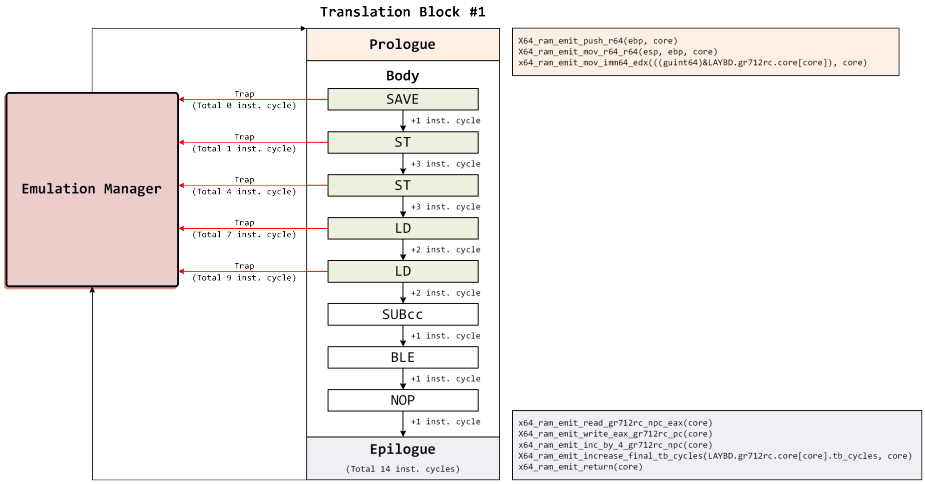
Cycle accurate emulation is an important requirements for space processors in satellite simulator. The interpreter-based emulator, `laysim-gr712rc`, can achieve cycle accuracy by modeling annul operations, register dependency, L1 cache hit/miss cycles, write buffer cycles, branch hit/miss cycles, and inter-processor effects. However, this approach results in significant performance degradation, and real-time performance cannot be guaranteed when the emulator is integrated into a satellite simulator.

Most dynamic binary translators (DBTs) do not provide instruction cycle counting mechanism due to the focus on achieving high performance. In a DBT system, the basic concept of instruction cycle counting is to accumulate instruction cycle counts as the basic block level, considering traps and branch conditions, whereas an interpreter updates instruction cycles for each instruction. A static cycle-approximate model can be applied in DBT systems to minimize performance penalties while maintaining acceptable cycle accuracy.

The instruction cycle counting method of `laysim-gr712rc-dbt` is illustrated in Figure 1-4. In this figure, five instructions are identified as trappable during the translation of a basic block. If a trap occurs during the execution of the final `LD` instruction, the emulator updates a

total of five instruction cycles instead of ten, as shown in this figure. This mechanism improves the cycle accuracy of `laysim-gr712rc-dbt` and constitutes a key feature of the emulator.

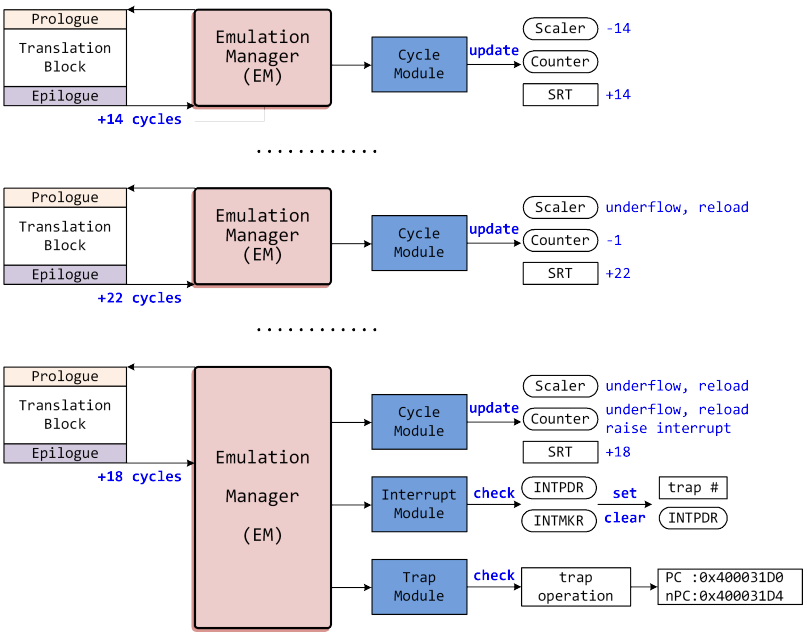
Figure 1-4. Instruction Cycle Counting in laysim-gr712rc-dbt



1.3.2 GPTIMER Emulation for Cycles

Cycle accuracy in processor emulation ensures that the emulator operates with precise timing. The simulated real time counter (SRT) is clocked according to the emulated processor clock and updated based on the number of executed instruction cycles. The SRT serves as the time source for emulated devices and simulated equipment, enabling synchronization with emulated processor.

Figure 1-5. GPTIMER Emulation in laysim-gr712rc-dbt



For example, the GR712RC processor includes a GPTIMER and a GRTIMER. GPTIMER provides four timers, while GRTIMER provides two timers. Each timer unit contains its own shared scaler register and timer count register.

The scaler register is clocked by the system clock (100MHz, 10nsec) and decrements on each cycle. When the scaler underflows, a tick pulse is generated for the timer, and the scaler is reloaded. The timer count register is decremented each time the scaler generates a tick pulse. When a timer underflows, a timer-specific interrupt is generated, and the counter is reloaded from reload register, as described in the GR712RC User's Manual.

In the GR712RC, the interrupt from GPTIMER-timer0 is used as the RTOS system clock, typically set to 100Hz for the RTEMS system clock. In `laysim-gr712rc-dbt`, the cycle module (depicted in Figure 1-3) maintains the simulation time of emulator. It updates the scaler and timer count registers based on the number of elapsed cycles in each translation block, and invokes timer interrupt upon expiration.

After a timer interrupt is raised, the interrupt module checks the interrupt conditions, manipulates the Multiprocessor Interrupt Controller (IRQMP), and sets the corresponding trap number, as shown in Figure 1-5. The trap module then handles the trap according to the processor's trap mechanism and updates the PC/nPC according to the trap base address. Finally, `laysim-gr712rc-dbt` either translates a new basic block when the TPC is not found in the SPC-to-TPC mape table, or executes the corresponding translation block when TPC is found.

1.3.3 Time Event Operation

The time event module in `laysim-gr712rc-dbt`, shown in Figure 1-3, is based on the Simulated Real Time (SRT) and provides services for timed-events in the event table. This mechanism is used to simulate the operation of various devices and equipment according to the behavior of the FSW operation.

The simulated devices and equipment can be activated by either software or hardware operation, and their activities must be synchronized with the SRT of `laysim-gr712rc-dbt`. The event table consists of four members;

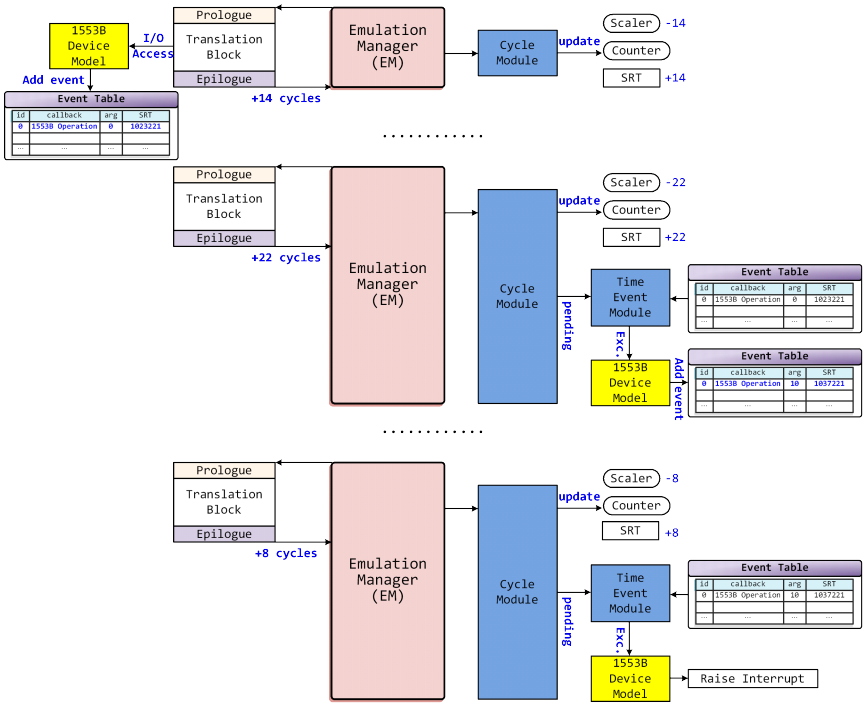
1. an event ID,
2. a callback function of the simulator,
3. an argument to be passed to the callback, and
4. the scheduled execution time based on the SRT.

When the FSW controls a device via APB operation, the corresponding simulation function is executed, and the next execution time is registered in the event table. After the execution of translation block, the cycle module of `laysim-gr712rc-dbt` updates the simulation time and checks pending events in the event table.

If a pending event exists, the time event module compares its scheduled execution time with the SRT. Once the scheduled time is reached, the registered callback function is executed. After execution, the simulation function registers the next execution time until the simulation operation is completed. During this process, it may raise a dedicated interrupt to signal the completion of the operation.

Figure 1-6 illustrates an example of timed operation for 1553B device emulation. This mechanism ensures that the simulation models remain synchronized with the SRT of `laysim-gr712rc-dbt`, thereby accurately emulating the behavior of the simulated devices and equipment.

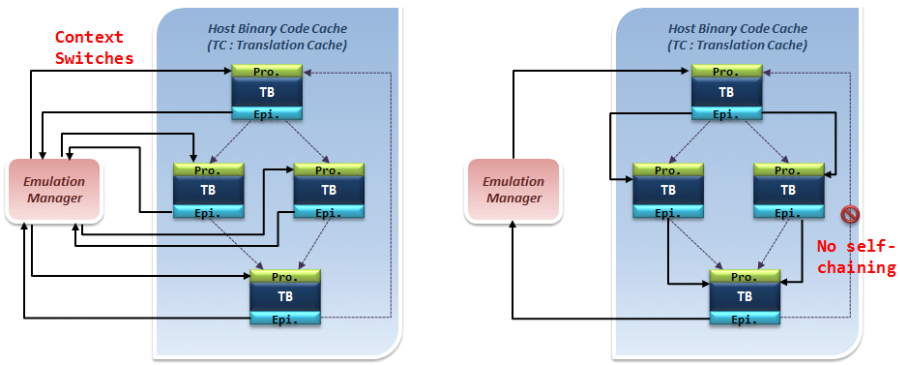
Figure 1-6. Timed Operation for 1553BRM Emulation in laysim-gr712rc-dbt



1.3.4 Translation Block Chaining

Translation Block (TB) chaining is a well-known technique to improve the performance of dynamic binary translation. Normally, when the execution of a TB reaches its end, control is returned to the emulation manager (EM). This context switching introduces significant overhead. By reducing the number of returns to the EM and allowing direct jumps between associated basic blocks, TB chaining provides substantial performance improvement, as illustrated in Figure 1-7.

Figure 1-7. Translation Block Chaining



`laysim-gr712rc-dbt` also adopts TB chaining to enhance performance significantly, but does not allow self-chaining. Self-chaining is restricted because it would prevent returning to the EM, except in the case of a trap condition.

Each TB consists of a prologue and an epilogue. The epilogue updates the PC/nPC, increments the total accumulated instruction cycles of the TB, and returns control to the EM. After TB execution, the EM checks whether the successor basic block has already been translated. If

it has not yet been translated, the EM modifies the epilogue of the predecessor TB to insert a jump to the next TB body, which is allocated by the translator and preceded by its prologue. If the successor TB has already been translated, the EM retrieves its TPC from the map table and establishes a direct link by overwriting the epilogue with a jump to the successor TB.

The EM of `laysim-gr712rc-dbt` prevents self-chaining by using a self-flagging mechanism in the map table. This ensures that internal infinite loops are avoided while still benefiting from performance gains through TB chaining.

In `laysim-gr712rc-dbt`, only direct branch instructions such as `BiCC`, `FBfCC` and `CALL` where the destination address does not change are eligible for chaining by the EM. On the other hand, indirect branch instructions in the SPARC architecture, of which only `JMPL` and `RETT` exists, simply return control to the EM without chaining.

Direct branch instructions are divided into two categories:

1. Unconditional branch instructions:

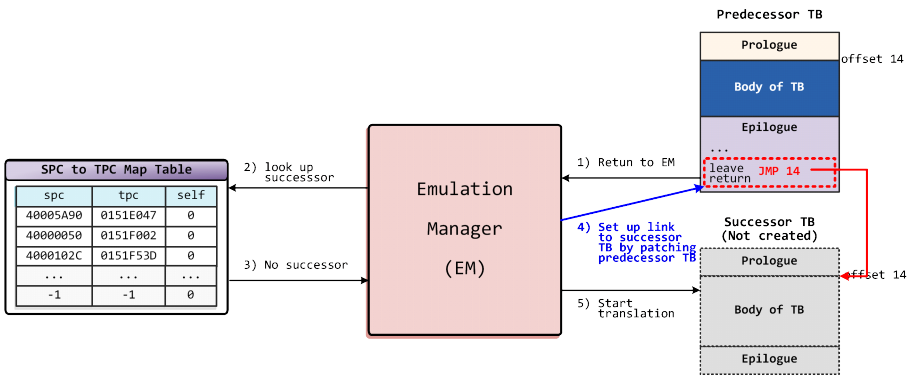
   Examples include `CALL`, `BA` (branch always without icc) in `BiCC`, and `FBA` (branch always without fcc), `FBN` (branch never without fcc) in `FBfCC`. These instructions only have a single fixed branch destination address.

2. Conditional branch instructions:

   These instructions can branch to one of two possible destination depending on the evaluation result of the condition. Therefore, the EM applies a different chaining strategy for these cases.

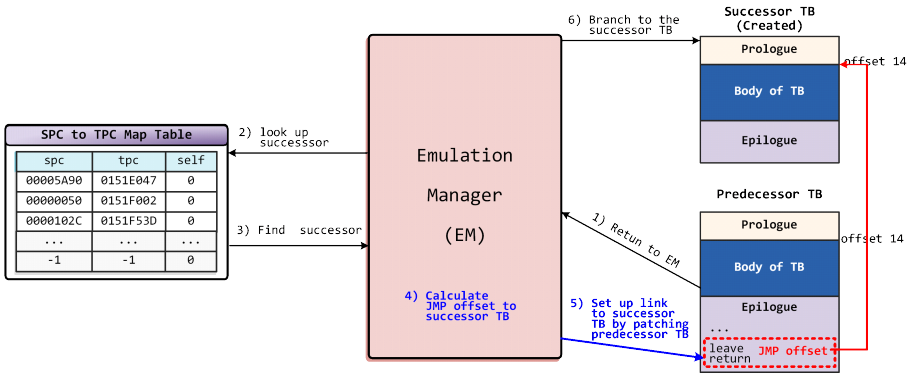
Figure 1-8 illustrates the chaining process for unconditional branches when a successor TB is not yet found in the SPC-to-TCP map table (i.e., the successor TB has not been translated). In this case, the EM patches the epilogue of the predecessor TB to insert a jump to the successor TB with a constant offset of 14, and then translates the successor basic block. When the predecessor TB is executed again, control flows directly to the successor TB without returning to the EM, thereby improving performance through TB chaining.

Figure 1-8. TB Chaining on Missing Successor for Unconditional Branch



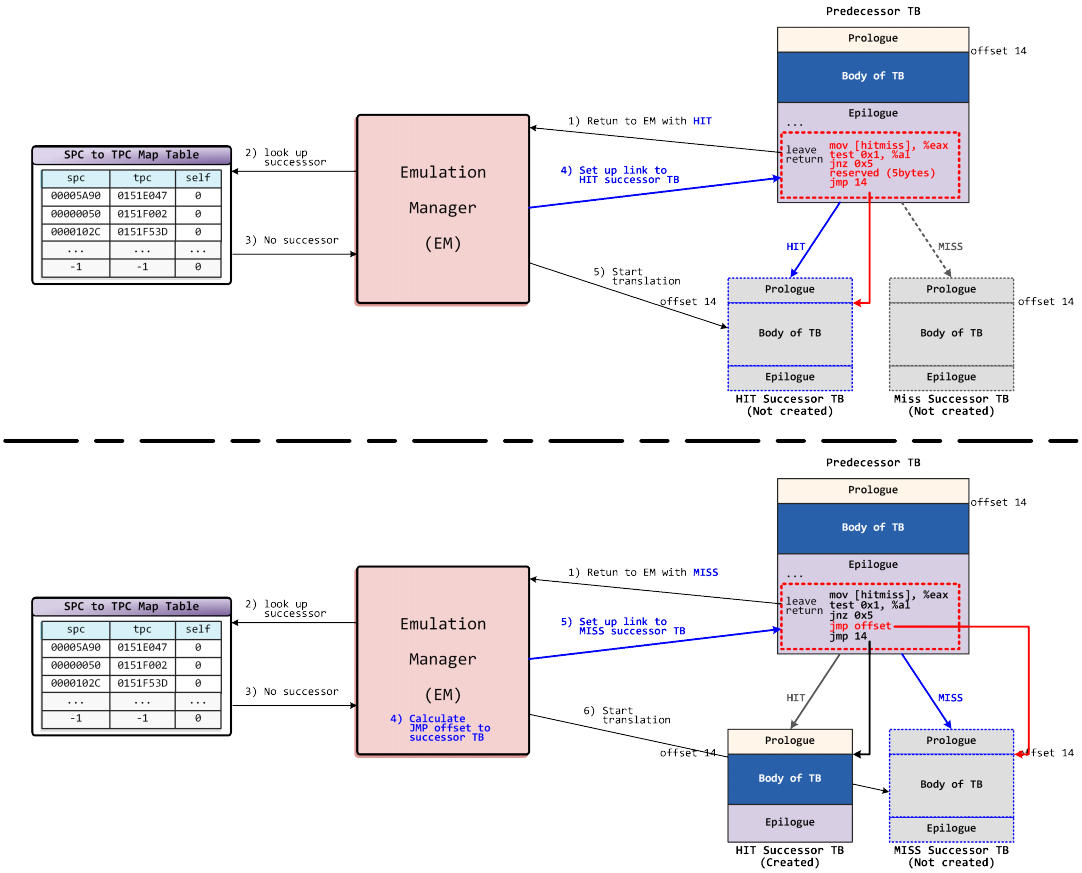
When the successor TB is already present in the SPC-to-TPC map table (i.e., the successor TB has been translated), the chaining step for an unconditional branch are shown in Figure 1-9. In this case, the EM overwrites the epilogue of the predecessor TB with a direct jump to the successor TB. Since the successor TB already exists in the translation cache, the EM can calculate the exact jump offset using the SPC-to-TPC map table. After updating the epilogue, control is transferred immediately to the successor TB for execution without returning to the EM.

Figure 1-9. TB Chaining on Finding Successor for Unconditional Branch



The steps of TB chaining for a conditional branch are more complex, as illustrated in Figure 1-10. When a TB returns to the EM due to a conditional branch hit, the EM consults the SPC-to-TPC map table to locate the TPC of the hit successor TB. If the hit TPC is not found in the map table, the EM patches the epilogue of the predecessor TB to link to the hit successor TB, which has not yet been translated.

Figure 1-10. TB Chaining for Conditional Branch



Unlike the unconditional branch, which requires only a single `jmp` instruction of x64, conditional branch chaining requires five x64 instructions. The first instruction moves the value of the hitmiss variable (updated during execution of the conditional branch in the predecessor TB) into the `eax` register. The second instruction tests whether the branch was taken using the `eax` register. The third instruction

is a `jnz` with a fixed offset of 5, which directs execution to the fifth instruction. The fourth instruction is reserved for the branch-miss case, since the miss successor TB cannot yet be identified. The fifth instruction is `jmp` to the hit successor TB, which will be translated next with a constant offset 14. After patching, the EM proceeds to translate the hit successor TB.

On subsequent execution, if the predecessor TB returns to the EM with a conditional branch miss, the EM again looks up the map table. If the miss TPC is not found, the EM links the miss successor TB by overwriting the reserved fourth instruction with a `jmp` to the miss successor TB using the calculated offset. The EM then begins translation of the miss successor TB.

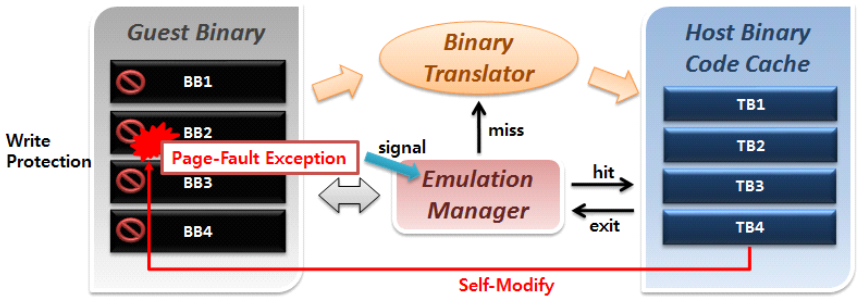
Once both the hit and miss successor TBs have been created, execution of the predecessor TB can directly transfer to either successor TB depending on the branch outcome, without returning to the EM.

1.3.5 Self-Modifying Code

Self-modifying code refers to guest code that modifies itself after being translated into a Translation Block (TB). This presents a potential issue for dynamic binary translation, because the code being executed is the translated TB, not the original guest code. As a result, the TB in the translation cache may no longer correspond to the modified guest code.

To address this issue, many DBT system marks the original guest code as write-protected at runtime using system calls such as `mprotect()` of Linux or `VirtualProtect()` of Windows, as shown in Figure 1-11. Any attempt to write to a protected page triggers a trap, sending a signal to the EM. Upon receiving this signal, all affected Translation Blocks, or only the TB corresponding to the modified page, are flushed from the translation cache. The EM then performs retranslation of the affected guest code.

Figure 1-11. General Handling Mechanism for Self-Modifying Code



1. Register signal handler for SIGSEGV
2. Set PAGE_READONLY to the basic block after translation using *VirtualProtect()*
3. Any attempt to write a page in the protection are will be trapped and send to signal to the EM
4. The EM flushes all translated code or the translation corresponding to the modified page
5. The EM starts retranslation

However, the use of `VirtualProtect()`/`mprotect()` can have side effect when employing a linear Translation Cache (TC). On x64 architecture, memory protection attributes can only be applied per page, with a minimum size of 4KB. In VxWorks 5.4, the guest RTOS used here, the kernel's code and data are mixed within the Task Control Block (TCB) area. Any modification of data in the TCB by VxWorks can mistakenly trigger retranslation, being falsely recognized as self-modifying code.

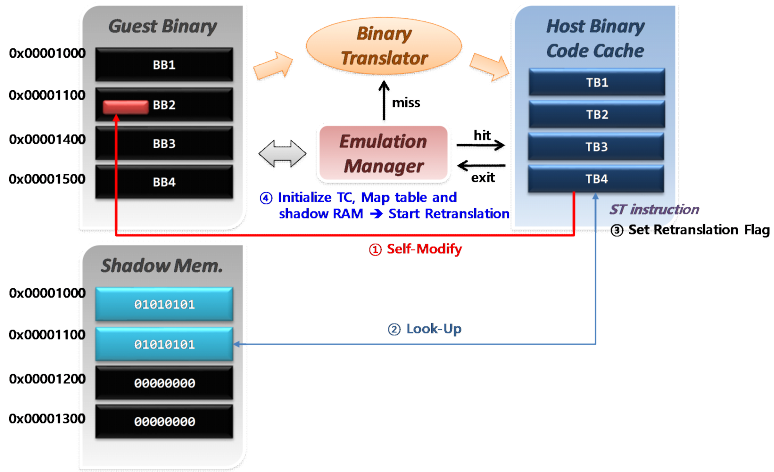
This issue is resolved by sing shadow memory, which mirrors the translated memory region without relying on system calls. By tracking

modifications in the shadow memory, `laysim-gr712rc-dbt` can correctly detect true self-modifying code while avoiding unnecessary retranslation.

To detect self-modifying code, `laysim-gr712rc-dbt` sets the corresponding address in the shadow memory during instruction translation. When a RAM write instruction is decoded, the DBT engine adds logic to check whether the target address has already been translated by consulting the shadow memory.

If a self-modifying case is detected, the EM reinitializes the TC, the SPC-to-TCP map table, and shadow memory, then begins retranslating the affected code. Figure 1-12 illustrates the overall workflow for handling self-modifying code in `laysim-gr712rc-dbt`.

Figure 1-12. Handling Mechanism for Self-Modifying Code in laysim-gr712rc-dbt

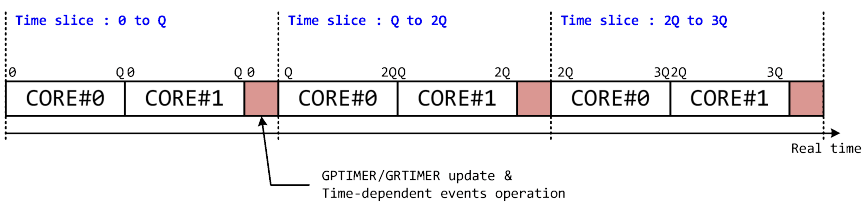


1.4 Multicore Scheduling in laysim-GR712RC

`laysim-gr712rc`, an interpreter-based emulator, schedules GR712RC multicore execution at the instruction and cycle level. The EM selects the core with the lowest local cycles, fetches an instruction, and executes it while updating the core's local cycles according to L1 cache hit/miss. The EM also updates GPTIMER/GRTIMER and processes time-dependent events registered by other units, such as APBUART or GRSpW2. This fine-grained scheduling ensures high cycle accuracy, but performance is significantly reduced in SMP applications.

`laysim-gr712rc-dbt` uses time quantum (TQ) scheduling for multicore execution and synchronization. Figure 1-13 illustrates the time quantum scheduling in `laysim-gr712rc-dbt`. By default, the time quantum is 200 cycles (2us) and can be modified using '`-quantum` VALUE' at startup.

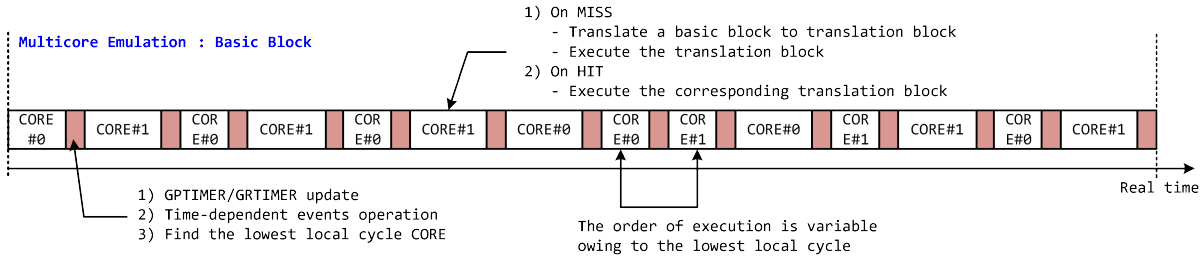
Figure 1-13. Time Quantum Scheduling of laysim-gr712rc-dbt



During multicore emulation, the EM executes CORE#0 and CORE#1 in round-robin order. After each quantum, it updates GPTIMER/GRTIMER (scaler and counter values, and raises GPTIMER/GRTIMER interrupts) and processes timed events. This time quantum thus defines the minimum synchronization interval between all cores. This method increases performance, but may introduce synchronization violations in strict real-time systems.

`laysim-gr712rc-dbt` allows switching multicore emulation from time-quantum scheduling to basic block scheduling, as shown in Figure 1-14. By setting '`-quantum` 0' at startup, the EM changes the multicore scheduling method to the basic block scheduling. In this mode, the EM selects the core with the lowest local cycles, executes its TB, updates GPTIMER/GRTIMER, and processes time-dependent events. Consequently, the minimum synchronization interval between all cores corresponds to the executed cycles of the TB. This method ensures strict synchronization across all cores but inevitably reduces real-time performance.

Figure 1-14. Basic Block Scheduling of laysim-gr712rc-dbt



1.5 Other Features in laysim-GR712RC

1.5.1 Emulation of AMBA Plug & Play Information

`laysim-GR712RC` provides the AMBA plug & play (P&P) information as implemented in the GR712RC processor. Figure 1-15 shows the P&P information of `laysim-GR712RC`. This information is used in the `BCC/RCC`, and `rtems-shell` example of showing emulated H/W devices of `laysim-GR712RC`.

Figure 1-15. AMBA P&P Information of laysim-GR712RC

Core	Function	Vendor ID	Device ID	Emulation in laysim-GR712RC
LEON3-FT	Fault-tolerant SPARC V8 Processor	0x01	0x053	Supported by laysim-GR712RC
DSU3	Multi-processor Debug support unit	0x01	0x004	Only P&P information
IRQMP	Multi-Processor Interrupt Controller	0x01	0x00D	Supported by laysim-GR712RC
APBCTRL	AMBA APB Bridge with Plug & Play	0x01	0x006	Controlled by laysim-GR712RC
FTMCTRL	8/32-bit memory controller with EDAC	0x01	0x054	Supported by laysim-GR712RC (No EDAC)
AHBSTAT	AHB Status Register	0x01	0x052	Supported by laysim-GR712RC
FTAHBRAM	Fault Tolerant On-Chip Memory	0x01	0x054	Supported by laysim-GR712RC (No code area)
AHBUART	Serial/AHB debug interface	0x01	0x007	Only P&P information
AHBJTAG	JTAG/AHB debug interface	0x01	0x01C	Only P&P information
GRSpW2	SpaceWire link with RMAP	0x01	0x029	Supported by laysim-GR712RC

B1553BRM	MIL-STD1553B BC/RT/BM Controller	0x01	0x072	Supported by laysim-GR712RC (BC/RT) RT mode is supported for EDISOFT RTEMS-Impr testsuites using loadable module
CANOC	OC CAN Controller	0x01	0x019	Supported by laysim-GR712RC
GRETH	10/100 Ethernet MAC with EDCL	0x01	0x01D	Supported by laysim-GR712RC
APBUART	Programmable UART with APB Interface	0x01	0x00C	Supported by laysim-GR712RC
GPTIMER	General Purpose Timer Unit	0x01	0x011	Supported by laysim-GR712RC
GRTIMER	General Purpose Timer Unit with latch	0x01	0x038	Supported by laysim-GR712RC
GRGPIO	General Purpose I/O port	0x01	0x01A	Supported by laysim-GR712RC
GRTC	CCSDS Telecommand Decoder	0x01	0x031	Not supported in laysim-GR712RC
GRTM	CCSDS Telemetry Encoder	0x01	0x030	Not supported in laysim-GR712RC
SATCAN	Proprietary	0x01	0x080	Not supported in laysim-GR712RC
CANMUX	CAN Bus Multiplexer	0x01	0x081	Not supported in laysim-GR712RC
CLKGATE	Clock gating module	0x01	0x02C	Supported by laysim-GR712RC
GRASCS	ASCS16 Controller	0x01	0x043	Not supported in laysim-GR712RC
GRSLINK	SLINK Controller	0x01	0x02F	Not supported in laysim-GR712RC
SPICTRL	SPI Controller (Master)	0x01	0x02D	Supported by laysim-GR712RC
I2CMST	I2C Master	0x01	0x08	Supported by laysim-GR712RC
GRGPREG	General Purpose Register	0x01	0x087	Not supported in laysim-GR712RC

1.5.2 Memory Map of laysim-GR712RC

laysim-GR712RC provides 8MB MRAM, 8MB SRAM as default and AHB/APB buses from GR712RC processor as shown in Figure 1-16. CPU Frequency, ROM size, RAM size, and APBUART#0 baudrate can be changed using start-up option.

Figure 1-16. Memory Map of laysim-GR712RC

Core		Address Range	Type	Bus	Emulation in laysim-GR712RC
FTMCTRL		0x00000000 - 0x20000000	PROM Area	AHB	• ELF/binary file can be loaded to ROM • 8MB ROM is implemented as default, but it can be changed using -romsz option
		0x20000000 - 0x40000000	I/O Area		Supported in laysim-GR712RC - Need loadable I/O module to extend
		0x40000000 - 0x80000000	SRAM/SDRAM Area		• ELF/AOUT/binary file can be loaded to SRAM • 8MB SRAM is implemented as default, but it can be changed using -ramsz option
APBBRIDGE0		0x80000000 - 0x80100000	APB Bridge	AHB	Supported by laysim-GR712RC
APBRID	FTMCTRL	0x80000000 - 0x80000100	Registers	APB	FTMCTRL is supported without EDAC
	APBUART0	0x80000100 - 0x80000200	Registers	APB	APBUART is supported with Tx/Rx FIFO - Need loadable UART module to extend
	IRQMP	0x80000200 - 0x80000300	Registers	APB	IRQMP is supported with extended interrupt and interrupt broadcast
	GPTIMER	0x80000300 - 0x80000400	Registers	APB	GPTIMER is supported
	SPICTRL	0x80000400 -	Registers	APB	Supported in laysim-GR712RC

		0x80000500			- Need loadable SPI module to extend
	CANMUX	0x80000500 - 0x80000600	Registers	APB	Dummy operation
	GRGPREG	0x80000600 - 0x80000700	Registers	APB	Dummy operation
	GRASCS	0x80000700 - 0x80000800	Registers	APB	Not supported in laysim-GR712RC
	GRSLINK	0x80000800 - 0x80000900	Registers	APB	Not supported in laysim-GR712RC
	GRGPIO0	0x80000900 - 0x80000A00	Registers	APB	GRGPIO is supported with interrupt generation - Need loadable GPIO module to extend
	GRGPIO1	0x80000A00 - 0x80000B00	Registers	APB	GRGPIO is supported with interrupt generation - Need loadable GPIO module to extend
	GRTM	0x80000B00 - 0x80000C00	Registers	APB	Not supported in laysim-GR712RC
	I2CMST	0x80000C00 - 0x80000D00	Registers	APB	Supported in laysim-GR712RC - Need loadable I2C module to extend
	CLKGATE	0x80000D00 - 0x80000E00	Registers	APB	CLKGATE is supported
	GRETH	0x80000E00 - 0x80000F00	Registers	APB	GRETH is supported
	AHBSTAT	0x80000F00 - 0x80001000	Registers	APB	AHBSTAT is supported
	APB0 P&P	0x800FF000 - 0x80100000	P&P Configuration	APB	P&P information is supported
APBBRIDGE1		0x80100000 - 0x80200000	APB Bridge	AHB	Controlled by laysim-GR712RC
	APBUART1	0x80100100 - 0x80100200	Registers	APB	APBUART is supported with Tx/Rx FIFO - Need loadable UART module to extend
	APBUART2	0x80100200 - 0x80100300	Registers	APB	APBUART is supported with Tx/Rx FIFO - Need loadable UART module to extend
	APBUART3	0x80100300 - 0x80100400	Registers	APB	APBUART is supported with Tx/Rx FIFO - Need loadable UART module to extend
	APBUART4	0x80100400 - 0x80100500	Registers	APB	APBUART is supported with Tx/Rx FIFO - Need loadable UART module to extend
	APBUART5	0x80100500 - 0x80100600	Registers	APB	APBUART is supported with Tx/Rx FIFO - Need loadable UART module to extend
	GRTIMER	0x80100600 - 0x80100700	Registers	APB	GPTIMER is supported with latching
	GRSpW2-0	0x80100800 - 0x80100900	Registers	APB	GRSpW2 is supported with RMAP and loadable SpW Module
	GRSpW2-1	0x80100900 - 0x80100A00	Registers	APB	GRSpW2 is supported with RMAP and loadable SpW Module
	GRSpW2-2	0x80100A00 - 0x80100B00	Registers	APB	GRSpW2 is supported with loadable SpW Module
	GRSpW2-3	0x80100B00 - 0x80100C00	Registers	APB	GRSpW2 is supported with loadable SpW Module
	GRSpW2-4	0x80100C00 - 0x80100D00	Registers	APB	GRSpW2 is supported with loadable SpW Module
	GRSpW2-5	0x80100D00 -	Registers	APB	GRSpW2 is supported with loadable SpW Module

		0x80100E00			
	APB1 P&P	0x801FF000 - 0x80200000	P&P Configuration	APB	P&P information is supported
DSU3		0x90000000 - 0xA0000000	Registers	AHB	N/A
FTAHBRAM		0xA0000000 - 0xA0100000	RAM Area Registers	AHB APB	192KB FTAHBRAM is implemented (Not for code area) - It can be used for GRSpW2, 1553BRM memory
B1553BRM		0xFFF00000 - 0xFFF01000	Registers	AHB	B1553BRM BC/RT mode is supported - Need loadable 1553B BC/RT module to extend
GRTC		0xFFF10000 - 0xFFF10100	Registers	AHB	Not supported in laysim-GR712RC
SATCAN		0xFFF20000 - 0xFFF20100	Registers	AHB	Not supported in laysim-GR712RC
CANOC		0xFFF30000 - 0xFFF31000	Registers	AHB	CANOC is supported with loadable CAN Module and CAN Bus
AHB P&P		0xFFFFF000 - 0xFFFFFFFF	P&P Configuration	AHB	P&P information is supported

1.5.3 Interrupts of laysim-GR712RC

laysim-GR712RC supports interrupts according to the GR712RC processor as shown in Figure 1-17. All interrupts are configured by the AMBA P&P information.

Figure 1-17. Interrupts of laysim-GR712RC

Peripheral	INT#	Function	Emulation in laysim-GR712RC
AHBSTAT	1	AHB bus error	If GRSpW2 or B1553BRM sets an invalid address in their registers for operation, then INT#1 will be raised.
APBUART0	2	UART Tx/Rx interrupt	Supported by laysim-GR712RC
GRGPIO0 GRGPIO1	1-15	External I/O interrupt (3&4 are GPIO only)	Supported by laysim-GR712RC
CANOC	5-6	OCCAN interrupt (core 1-2)	Supported by laysim-GR712RC
GRTIMER	7	GRTIMER timer undeflow interrupt	Supported by laysim-GR712RC
GPTIMER	8-11	GPTIMER timer underflow interrupt	Supported by laysim-GR712RC
IRQMP	12	IRQMP extended interrupt	Supported by laysim-GR712RC
SPICTRL, SLINK	13	SPI and SLINK interrupt	SPICTRL is supported by laysim-GR712RC SLINK is not supported by laysim-GR712RC
B1553BRM, GRETH, GRTC	14	1553, Ethernet and Telecommand interrupt	B1553BRM and GRETH are supported by laysim-GR712RC GRTC is not supported by laysim-GR712RC
ASCS	16	ASCS interrupt	ASCS is not supported by laysim-GR712RC
APBUART1-5	17-21	UART2-6 Rx/Tx interrupt	Supported by laysim-GR712RC
GRSpW2 - 0 to 5	22-27	SpaceWire interrupts	Supported by laysim-GR712RC
I2CMST	28	I2C master interrupt	Supported by laysim-GR712RC
GRTM	29	Telemetry encoder interrupt	GRTM is not supported by laysim-GR712RC
	30	Telemetry encoder time strobe interrupt	

1.6 Limitation of laysim-GR712RC Evaluation Version

The evaluation version of laysim-GR712RC does not impose execution limitations, except that users must acquire a 1-year free license as described in Chapter 2.2. The user-defined modules such as UART, I2C, SPI, GPIO, I/O, SpW nodes, 1553B RTs, and CAN nodes can be supported according to the separate license agreement. In the evaluation version, only pre-built loadable UART, I2C, SPI, GPIO, I/O,

external SpW nodes, 1553B RTs, and CAN nodes modules are provided.

1.7 Applicable Documents

[laysim-GR712RC-001] laysim-GR712RC Dual Core Processor Emulator - User's Manual

[laysim-GR712RC-002] GR712RC Examples from BCC/RCC/RTEMS5 Result on laysim-GR712RC

[laysim-GR712RC-003] Debugging with GDB on laysim-GR712RC

[laysim-GR712RC-004] Code coverage operation on laysim-GR712RC

[laysim-GR712RC-005] laysim-GR712RC User Loadable Modules - User's Manual (Only for professional ver.)

[laysim-GR712RC-006] laysim-GR712RC Shared Library - User's Manual

[laysim-GR712RC-007] GR712RC Synchronous Trap Test on laysim-GR712RC

[laysim-GR712RC-008] AIR 5.6.0 Test Result with laysim-GR712RC

[laysim-GR712RC-009] Debugging AIR 5.6.0 on laysim-GR712RC

[laysim-GR712RC-010] EDISOFT RETEMS4.8-Improvement Test Result with laysim-GR712RC

[laysim-GR712RC-011] Establish SDE for GR712RC in Linux

[laysim-GR712RC-012] RTEMS6.1 Build and Test on laysim-GR712RC

[laysim-GR712RC-013] CANOC Operation on laysim-GR712RC (Only for professional ver.)

[laysim-GR712RC-014] 1553BRM Operation on laysim-GR712RC (Only for professional ver.)

[laysim-GR712RC-015] GRSpW2 Operation on laysim-GR712RC (Only for professional ver.)

2. Installation

2.1 Installation for Linux x64

2.1.1 Preparation

In order to compile examples of `BCC-2.2.1`/`RCC-1.3.1` in Linux 64bit platform, 5 components should be installed with designated directories as shown in Figure 2-1. It is recommended to install them sequentially from top to bottom. Optionally, `RTEMS6.1` from git can be installed.

Figure 2-1. Install Cross Tool Chains for GR712RC in Linux

Component	Directory	File	Remark
BCC-2.2.1	/opt/bcc-2.2.1-gcc	bcc-2.2.1-gcc-linux64.tar.xz	
BCC-2.2.1 source code	/opt/bcc-2.2.1-gcc/src/bcc-2.2.1	bcc-2.2.1-src.tar.bz2	debugging BCC2 codes
RCC-1.3.1	/opt/rcc-1.3.1-gcc	sparc-rtems-5-gcc-10.2.0-1.3.1-linux.txz	-
RCC-1.3.1 source code	/opt/rcc-1.3.1-gcc/src/rcc-1.3.1	rtems-5-1.3.1-src.txz	debugging RCC-1.3.1 codes
MKPROM2	/opt/mkprom2	mkprom2-2.0.65.tar.gz	create boot-image

After installation, the toolchain executables can be invoked from the command prompt by adding the executable directory to the `PATH` environment variable in `.bashrc`.

2.1.2 Installation of laysim-GR712RC

`laysim-GR712RC` was built with gtk+2.0 (64bit) on CentOS 7, so there are no additional software packages for CentOS 7. But it is necessary to install `libcanberra-gtk-module` on CentOS 8 and Ubuntu 18.04 to suppress warning of 'Failed to load module "canberra-gtk-module". In case of Debian 10.3, `libgtk2.0` and `libcanberra-gtk-module` should be installed.

2.1.2.1 Installation of additional packages for CentOS 7

`laysim-GR712RC` was built with gtk+2.0 (64bit) on CentOS 7, and CentOS 7 x64 provides `gtk+2.0-2.24.31` as default when it selected GNOME Desktop during installation. So there are no additional software packages on CentOS 7 for `laysim-GR712RC`.

2.1.2.2 Installation of additional packages for CentOS 8 and Ubuntu 18.04

CentOS 8 x64 provides `gtk+2.0-2.24.31` as default when it selected GNOME Desktop during installation, and Ubuntu 18.04 provides `gtk+2.0-2.24.32-1` as default. But it is necessary to install `libcanberra-gtk-module` to suppress warning as shown in Figure 2-2.

Figure 2-2. Install libcanberra-gtk-module on Ubuntu 18.04

```
layright@ubuntu:~$ sudo apt-get install libcanberra-gtk-module
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
```

```
  libcanberra-gtk-module libcanberra-gtk0
0 upgraded, 2 newly installed, 0 to remove and 280 not upgraded.
Need to get 17.9 kB of archives.
After this operation, 84.0 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu bionic/universe amd64 libcanberra-gtk0 amd64 0.30-5ubuntu1 [7,864 B]
Get:2 http://us.archive.ubuntu.com/ubuntu bionic/universe amd64 libcanberra-gtk-module amd64 0.30-5ubuntu1 [10.0 kB]
Fetched 17.9 kB in 1s (16.1 kB/s)
Selecting previously unselected package libcanberra-gtk0:amd64.
(Reading database ... 140366 files and directories currently installed.)
Preparing to unpack .../libcanberra-gtk0_0.30-5ubuntu1_amd64.deb ...
Unpacking libcanberra-gtk0:amd64 (0.30-5ubuntu1) ...
Selecting previously unselected package libcanberra-gtk-module:amd64.
Preparing to unpack .../libcanberra-gtk-module_0.30-5ubuntu1_amd64.deb ...
Unpacking libcanberra-gtk-module:amd64 (0.30-5ubuntu1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Setting up libcanberra-gtk0:amd64 (0.30-5ubuntu1) ...
Setting up libcanberra-gtk-module:amd64 (0.30-5ubuntu1) ...
Processing triggers for libc-bin (2.27-3ubuntu1) ...
```

2.1.2.3 Installation of additional packages for Debian 10.3

Debian 10.3 x64 does not provide gtk+2.0 as default, so it is required to install `libgtk2.0` and `libcanberra-gtk-module` as shown in Figure 2-3.

Figure 2-3. Install libgtk2.0 and libcanberra-gtk-module on Debian 10.3

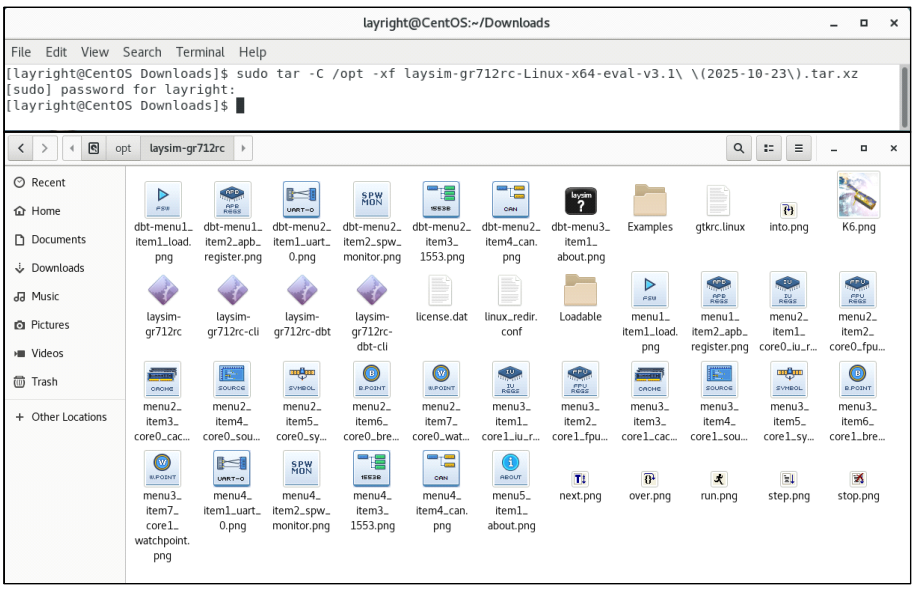
```
layright@debian:~$ sudo apt-get install libgtk2.0
[sudo] password for layright:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'libgtk2.0-bin' for regex 'libgtk2.0'
Note, selecting 'libgtk2.0-0' for regex 'libgtk2.0'
Note, selecting 'libgtk2.0-common' for regex 'libgtk2.0'
The following additional packages will be installed:
  libgail-common libgail18
The following NEW packages will be installed:
  libgail-common libgail18 libgtk2.0-0 libgtk2.0-bin libgtk2.0-common
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/4,767 kB of archives.
After this operation, 27.0 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 cdrom://[Debian GNU/Linux 10.3.0 _Buster_ - Official amd64 DVD Binary-1 20200208-12:08] buster/main amd64 libgtk2.0-common all 2.24.32-3 [2,698
kB]
Get:2 cdrom://[Debian GNU/Linux 10.3.0 _Buster_ - Official amd64 DVD Binary-1 20200208-12:08] buster/main amd64 libgtk2.0-0 amd64 2.24.32-3 [1,809 kB]
Get:3 cdrom://[Debian GNU/Linux 10.3.0 _Buster_ - Official amd64 DVD Binary-1 20200208-12:08] buster/main amd64 libgail18 amd64 2.24.32-3 [55.5 kB]
Get:4 cdrom://[Debian GNU/Linux 10.3.0 _Buster_ - Official amd64 DVD Binary-1 20200208-12:08] buster/main amd64 libgail-common amd64 2.24.32-3 [156 k
B]
Get:5 cdrom://[Debian GNU/Linux 10.3.0 _Buster_ - Official amd64 DVD Binary-1 20200208-12:08] buster/main amd64 libgtk2.0-bin amd64 2.24.32-3 [48.5 k
B]
Selecting previously unselected package libgtk2.0-common.
(Reading database ... 132907 files and directories currently installed.)
Preparing to unpack .../libgtk2.0-common_2.24.32-3_all.deb ...
Unpacking libgtk2.0-common (2.24.32-3) ...
Selecting previously unselected package libgtk2.0-0:amd64.
Preparing to unpack .../libgtk2.0-0_2.24.32-3_amd64.deb ...
Unpacking libgtk2.0-0:amd64 (2.24.32-3) ...
Selecting previously unselected package libgail18:amd64.
Preparing to unpack .../libgail18_2.24.32-3_amd64.deb ...
Unpacking libgail18:amd64 (2.24.32-3) ...
Selecting previously unselected package libgail-common:amd64.
Preparing to unpack .../libgail-common_2.24.32-3_amd64.deb ...
Unpacking libgail-common:amd64 (2.24.32-3) ...
Selecting previously unselected package libgtk2.0-bin.
Preparing to unpack .../libgtk2.0-bin_2.24.32-3_amd64.deb ...
Unpacking libgtk2.0-bin (2.24.32-3) ...
Setting up libgtk2.0-common (2.24.32-3) ...
Setting up libgtk2.0-0:amd64 (2.24.32-3) ...
Setting up libgail18:amd64 (2.24.32-3) ...
Setting up libgtk2.0-bin (2.24.32-3) ...
Setting up libgail-common:amd64 (2.24.32-3) ...
Processing triggers for man-db (2.8.5-2) ...
Processing triggers for libc-bin (2.28-10) ...

layright@debian:~$ sudo apt-get install libcanberra-gtk-module
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libcanberra-gtk0
The following NEW packages will be installed:
  libcanberra-gtk-module libcanberra-gtk0
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/27.0 kB of archives.
After this operation, 92.2 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 cdrom://[Debian GNU/Linux 10.3.0 _Buster_ - Official amd64 DVD Binary-1 20200208-12:08] buster/main amd64 libcanberra-gtk0 amd64 0.30-7 [12.5 k
B]
Get:2 cdrom://[Debian GNU/Linux 10.3.0 _Buster_ - Official amd64 DVD Binary-1 20200208-12:08] buster/main amd64 libcanberra-gtk-module amd64 0.30-7 [1
4.5 kB]
```

```
Selecting previously unselected package libcanberra-gtk0:amd64.
(Reading database ... 133183 files and directories currently installed.)
Preparing to unpack .../libcanberra-gtk0_0.30-7_amd64.deb ...
Unpacking libcanberra-gtk0:amd64 (0.30-7) ...
Selecting previously unselected package libcanberra-gtk-module:amd64.
Preparing to unpack .../libcanberra-gtk-module_0.30-7_amd64.deb ...
Unpacking libcanberra-gtk-module:amd64 (0.30-7) ...
Setting up libcanberra-gtk0:amd64 (0.30-7) ...
Setting up libcanberra-gtk-module:amd64 (0.30-7) ...
Processing triggers for libc-bin (2.28-10) ...
```

2.1.2.4 Installation of laysim-GR712RC

`laysim-GR712RC` should be installed to `/opt/laysim-gr712rc` as shown in Figure 2-4, and it is needed to add execution path in `.bashrc`.

Figure 2-4. Install laysim-GR712RC



The contents of `laysim-GR712RC` consist as below Figure 2-5.

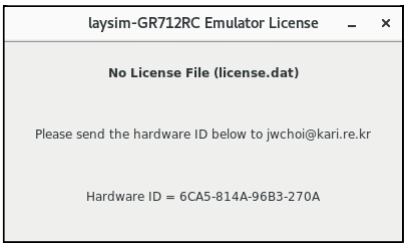
Figure 2-5. Contents of laysim-GR712RC

Directory	Files	Description
/opt/laysim-gr712rc	laysim-gr712rc	GR712RC dual core processor emulator based on interpretation
	laysim-gr712rc-cli	CLI version of laysim-gr712rc
	laysim-gr712rc-dbt	GR712RC dual core processor emulator based on DBT
	laysim-gr712rc-dbt-cli	CLI version of laysim-gr712rc-dbt
./Examples	/BCC-2.2.1	BCC2 example files from BCC2 distribution
	/RCC-1.3.1	RCC example files from RCC-1.3.1 distribution
	/XtratuM	XtratuM 4.6 r1 for GR712RC-SMP
	/Linux	Linux SMP for GR712RC
	/VxWorks7	WindRiver VxWorks 7 SMP for GR712RC
	/user-io	user IO examples - These examples can be used with loadable IO module
./Loadable	/loadable-1553RT-Modules	Pre-built loadable 1553B RT module

	/loadable-CAN-Nodes	Pre-built loadable CAN node module
	/loadable-GPIO-Modules	Pre-built loadable GPIO module
	/loadable-IO-Modules	Pre-built loadable I/O module
	/loadable-I2C-Modules	Pre-built loadable I2C module
	/loadable-SPI-Modules	Pre-built loadable SPI module
	/loadable-SpW-Nodes	Pre-built loadable SpW node module
	/loadable-UART-Modules	Pre-built loadable UART module

On first execution of `laysim-GR712RC`, it would request the license file as shown in Figure 2-6. The user should send the hardware ID to `jwchoi@kari.re.kr` for acquiring the license file. After the reception of the license file (`license.dat`), then the user copies it to `/opt/laysim-gr712rc`.

Figure 2-6. Need License File



2.2 Installation for Windows

2.2.1 Preparation

In order to compile `BCC-2.2.1`/`RCC-1.3.1` examples in Windows 11 64bit platform, 5 components should be installed with designated directories as shown in Figure 2-7. It is recommended to install them sequentially from top to bottom and to use the 7zip extractor. Optionally, `RTEMS6.1` from git can be installed.

Figure 2-7. Install Cross Tool Chains for GR712RC in Windows

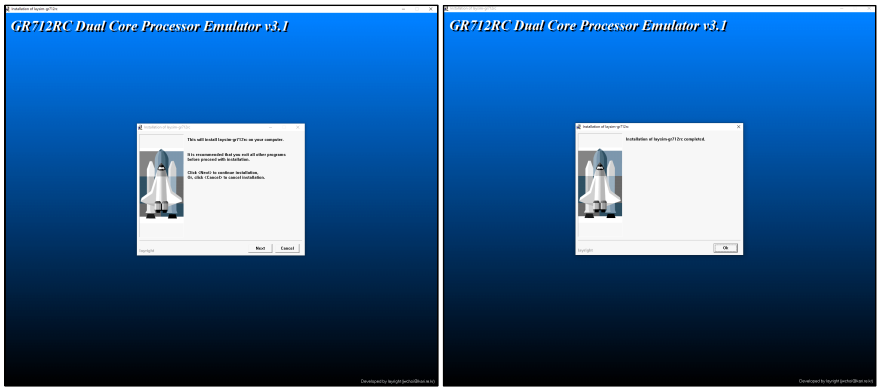
Component	Directory	File	Remark
BCC-2.2.1	C:\opt\bcc-2.2.1-gcc	bcc-2.2.1-gcc-mingw64.zip	
BCC-2.2.1 source code	C:\opt\bcc-2.2.1-gcc\src\bcc-2.2.1	bcc-2.2.1-src.tar.bz2	debugging BCC2 codes
RCC-1.3.1	C:\opt\rcc-1.3.1-gcc	sparc-rtems-5-gcc-10.2.0-1.3.1-mingw.zip	-
RCC-1.3.1 source code	C:\opt\rcc-1.3.1-gcc\src\rcc-1.3.1	rtems-5-1.3.1-src.txz	debugging RCC-1.3.1 codes
MKPROM2	C:\opt\mkprom2	mkprom2-2.0.65-mingw.tar.gz	create boot-image

After installation, the toolchain executables can be invoked from the command prompt by adding the executable directory to the PATH envir onment variable. (`C:\opt\bcc-2.2.1-gcc\bin;C:\opt\rcc-1.3.1-gcc\bin;C:\opt\mkprom2`)

2.2.2 Installation of laysim-GR712RC

Installation procedure of `laysim-GR712RC` is started by executing the installer, `laysim-GR712RC-installer.exe`. The installer will install `laysim-GR712RC` to `C:\KOMPSAT-FSS\laysim-gr712rc`, and `laysim-gr712rc` and `laysim-gr712rc-dbt` emulators are registered in Windows Programs.

Figure 2-8. Install laysim-GR712RC in Windows



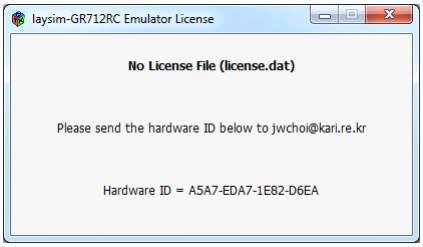
The contents of `laysim-GR712RC` consist as below Figure 2-9. MinGW or MSYS2 may be needed to build examples of `laysim-GR712RC`.

Figure 2-9. Contents of laysim-GR712RC

Directory	Files	Description
C:\KOMPSAT-FSS\laysim-gr712rc	laysim-gr712rc.exe	GR712RC dual core processor emulator based on interpretation
	laysim-gr712rc-cli.exe	CLI version of laysim-gr712rc
	laysim-gr712rc-dbt.exe	GR712RC dual core processor emulator based on DBT
	laysim-gr712rc-dbt-cli.exe	CLI version of laysim-gr712rc-dbt
./Examples	/BCC-2.2.1	BCC2 example files from BCC2 distribution
	/RCC-1.3.1	RCC example files from RCC-1.3.1 distribution
	/XtratuM	XtratuM 4.6 r1 for GR712RC-SMP
	/Linux	Linux SMP for GR712RC
	/VxWorks7	WindRiver VxWorks 7 SMP for GR712RC
	/user-io	user IO examples - These examples can be used with loadable IO module
./Loadable	/loadable-1553RT-Modules	Pre-built loadable 1553B RT module
	/loadable-CAN-Nodes	Pre-built loadable CAN node module
	/loadable-GPIO-Modules	Pre-built loadable GPIO module
	/loadable-IO-Modules	Pre-built loadable I/O module
	/loadable-I2C-Modules	Pre-built loadable I2C module
	/loadable-SPI-Modules	Pre-built loadable SPI module
	/loadable-SpW-Nodes	Pre-built loadable SpW node module
	/loadable-UART-Modules	Pre-built loadable UART module

On first execution of `laysim-GR712RC`, it would request the license file as shown in Figure 2-10. The user should send the hardware ID to `jwchoi@kari.re.kr` for acquiring the license file. After the reception of the license file (`license.dat`), then the user copies it to `C:\KOMPSAT-FSS\laysim-gr712rc`.

Figure 2-10. Need License File



2.3 Compilation of Examples

Once the `BCC-2.2.1`, `RCC-1.3.1` and `MKPROM2` are installed, examples provided by `laysim-GR712RC` can be compiled and generated to PROM binary image as shown in Figure 2-11 and 2-12.

Figure 2-11. Build BCC-2.2.1 Examples

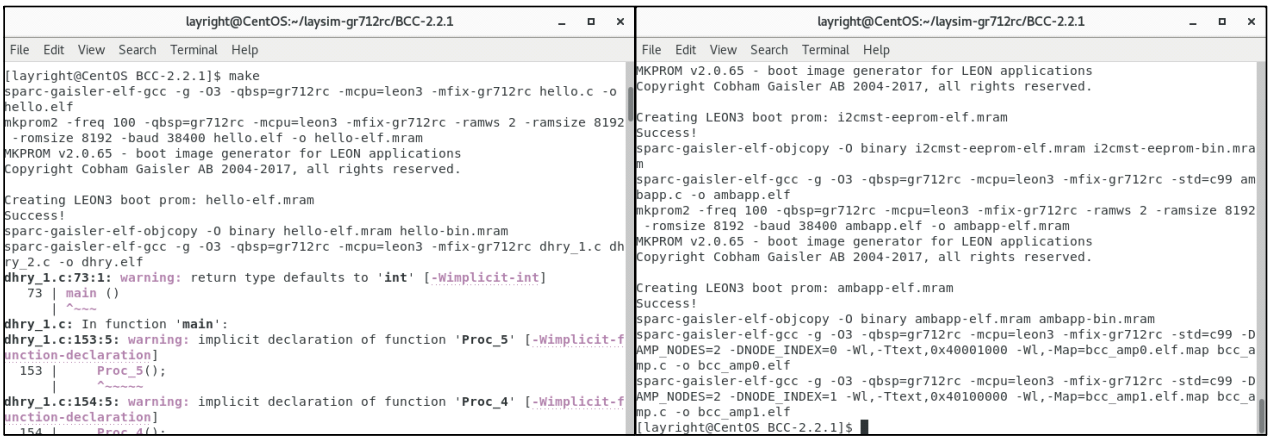


Figure 2-12. Build RCC-1.3.1 Samples

3. laysim-GR712RC Operation

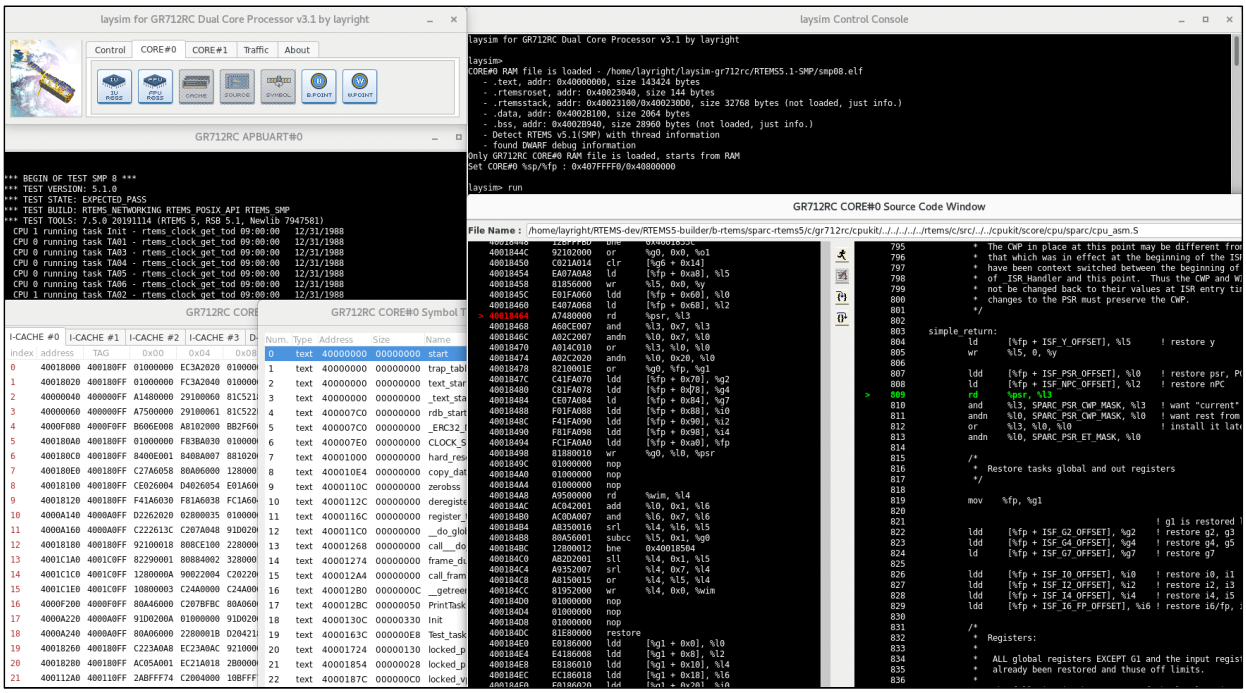
3.1 Emulators of laysim-GR712RC

`laysim-GR712RC` provides four types of emulators: `laysim-gr712rc`/`laysim-gr712rc-cli` and `laysim-gr712rc-dbt`/`laysim-gr712rc-dbt-cli`.

3.1.1 laysim-gr712rc

`laysim-gr712rc` is a GUI-based, cycle-accurate emulator for the GR712RC dual core processor using an interpretation method. It includes an embedded source-level debugger and supports the MMU of GR712RC. Figure 3-1 shows the overall execution of `laysim-gr712rc`.

Figure 3-1. laysim-gr712rc Interfaces



3.1.2 laysim-gr712rc-cli

`laysim-gr712rc-cli` is the CLI version of `laysim-gr712rc`. It provides the same functionality of `laysim-gr712rc` without a GUI, and it supports the debugging with GDB. `laysim-gr712rc-cli` only supports for the code coverage in the same format as `TSIM3-LEON3`. '`-cov`' start-up option should be enabled for the code coverage.

Figure 3-2. laysim-gr712rc-cli Interfaces

laysim-gr712rc-cli in Linux

```
[layright@CentOS RTEMS5.1-SMP]$ laysim-gr712rc-cli -core0 smp08.elf
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - smp08.elf
   - .text, addr: 0x40000000, size 143424 bytes
   - .rtemsroset, addr: 0x40023040, size 144 bytes
   - .rtemsstack, addr: 0x40023100/0x400230D0, size 32768 bytes (not loaded, just inf
o.)
   - .data, addr: 0x4002B100, size 2064 bytes
   - .bss, addr: 0x4002B940, size 28960 bytes (not loaded, just info.)
```

laysim-gr712rc-cli in Windows

```
PS C:\laysim-gr712rc\RTEMS5.1-SMP> laysim-gr712rc-cli -core0 smp08.elf
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - smp08.elf
   - .text, addr: 0x40000000, size 143424 bytes
   - .rtemsroset, addr: 0x40023040, size 144 bytes
   - .rtemsstack, addr: 0x40023100/0x400230D0, size 32768 bytes (not loaded, just inf
o.)
   - .data, addr: 0x4002B100, size 2064 bytes
   - .bss, addr: 0x4002B940, size 28960 bytes (not loaded, just info.)
```

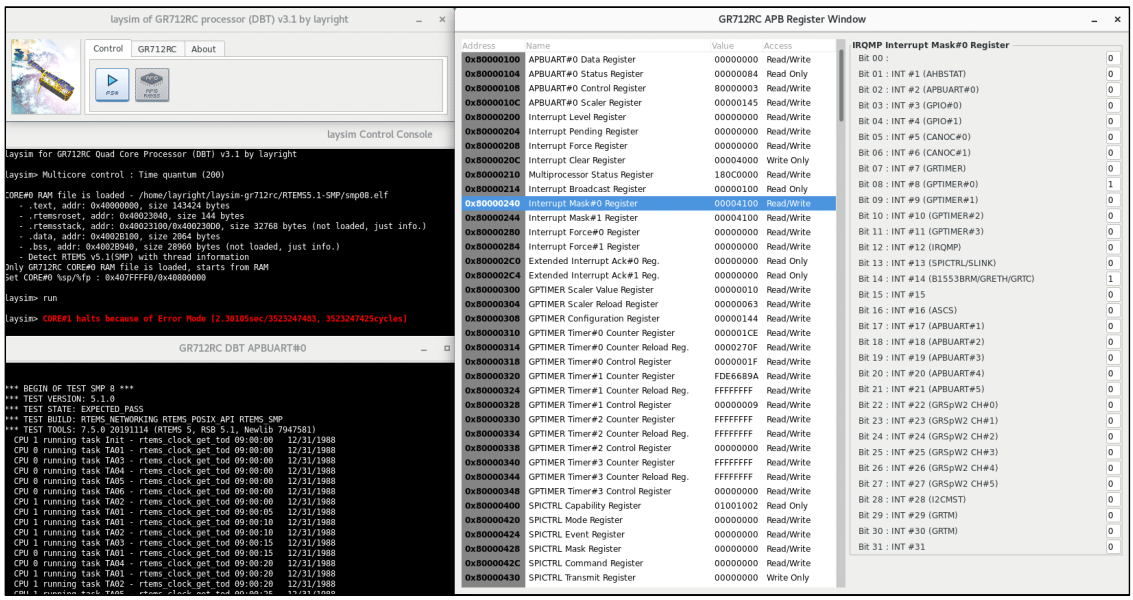
```
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
laysim> run

*** BEGIN OF TEST SMP 8 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
  CPU 1 running task Init - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA03 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA04 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA05 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA06 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:05   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA03 - rtems_clock_get_tod 09:00:15   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:15   12/31/1988
  CPU 0 running task TA04 - rtems_clock_get_tod 09:00:20   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:20   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:20   12/31/1988
  CPU 1 running task TA05 - rtems_clock_get_tod 09:00:25   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:25   12/31/1988
  CPU 0 running task TA03 - rtems_clock_get_tod 09:00:30   12/31/1988
  CPU 0 running task TA06 - rtems_clock_get_tod 09:00:30   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:30   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:30   12/31/1988

*** END OF TEST SMP 8 ***

CORE#1 halts because of Error Mode [3523939858cycles/2.01317sec]
laysim> perf

 Performance statistics for CORE#0
   Cycles       : 3523939858
   Instructions : 11028335
   Overall CPI  : 2.36
   CORE#0 performance (100MHz) : 42.40 MOPS (42.40 MIPS, 0.00 MFLOPS)
   Simulated time              : 35.24 s
   Processor utilization       : 0.74 %

 Performance statistics for CORE#1
   Cycles       : 3523939858
   Instructions : 24049709
   Overall CPI  : 2.09
   CORE#1 performance (100MHz) : 47.94 MOPS (47.94 MIPS, 0.00 MFLOPS)
   Simulated time              : 35.24 s
   Processor utilization       : 1.42 %

 Performance of laysim-gr712rc
   Real-time performance       : 1750.44 %
   laysim-gr712rc performance  : 17.42 MIPS
   Wall clock                  : 2.01 s
laysim>
```

```
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
laysim> run

*** BEGIN OF TEST SMP 8 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
  CPU 1 running task Init - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA03 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA04 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA05 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA06 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:05   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA03 - rtems_clock_get_tod 09:00:15   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:15   12/31/1988
  CPU 0 running task TA04 - rtems_clock_get_tod 09:00:20   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:20   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:20   12/31/1988
  CPU 1 running task TA05 - rtems_clock_get_tod 09:00:25   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:25   12/31/1988
  CPU 0 running task TA03 - rtems_clock_get_tod 09:00:30   12/31/1988
  CPU 0 running task TA06 - rtems_clock_get_tod 09:00:30   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:30   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:30   12/31/1988

*** END OF TEST SMP 8 ***

CORE#1 halts because of Error Mode [3523939858cycles/2.01393sec]
laysim> perf

 Performance statistics for CORE#0
   Cycles       : 3523939858
   Instructions : 11028335
   Overall CPI  : 2.36
   CORE#0 performance (100MHz) : 42.40 MOPS (42.40 MIPS, 0.00 MFLOPS)
   Simulated time              : 35.24 s
   Processor utilization       : 0.74 %

 Performance statistics for CORE#1
   Cycles       : 3523939858
   Instructions : 24049709
   Overall CPI  : 2.09
   CORE#1 performance (100MHz) : 47.94 MOPS (47.94 MIPS, 0.00 MFLOPS)
   Simulated time              : 35.24 s
   Processor utilization       : 1.42 %

 Performance of laysim-gr712rc
   Real-time performance       : 1749.78 %
   laysim-gr712rc performance  : 17.42 MIPS
   Wall clock                  : 2.01 s
laysim>
```

3.1.3 laysim-gr712rc-dbt

`laysim-gr712rc-dbt` is a GUI-based high-performance, cycle-approximate processor emulator that uses dynamic binary translation (DBT). It provides similar interfaces of `laysim-gr712rc` without source-level debugger and L1 cache for speed-up. Figure 3-3 shows the overall execution of `laysim-gr712rc-dbt`.

Figure 3-3. laysim-gr712rc-dbt Interfaces



3.1.4 laysim-gr712rc-dbt-cli

`laysim-gr712rc-dbt-cli` is the CLI version of `laysim-gr712rc-dbt`. It provides the same functionality of `laysim-gr712rc-dbt` without a GUI, and it supports the debugging with GDB.

Figure 3-4. laysim-gr712rc-dbt-cli Interfaces

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS RTEMS5.1-SMP]$ laysim-gr712rc-dbt-cli -core0 smp08.elf
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - smp08.elf
   - .text, addr: 0x40000000, size 143424 bytes
   - .rtemsroset, addr: 0x40023040, size 144 bytes
   - .rtemsstack, addr: 0x40023100/0x400230D0, size 32768 bytes (not loaded, just inf
o.)
   - .data, addr: 0x4002B100, size 2064 bytes
   - .bss, addr: 0x4002B940, size 28960 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
laysim> run

*** BEGIN OF TEST SMP 8 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
  CPU 1 running task Init - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA03 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA04 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA05 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA06 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:05   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA03 - rtems_clock_get_tod 09:00:15   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:15   12/31/1988
  CPU 0 running task TA04 - rtems_clock_get_tod 09:00:20   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:20   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:20   12/31/1988
  CPU 1 running task TA05 - rtems_clock_get_tod 09:00:25   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:25   12/31/1988
  CPU 0 running task TA03 - rtems_clock_get_tod 09:00:30   12/31/1988
  CPU 0 running task TA06 - rtems_clock_get_tod 09:00:30   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:30   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:30   12/31/1988

*** END OF TEST SMP 8 ***

CORE#1 halts because of Error Mode [0.284744sec/3523247483, 3523247425cycles]
laysim> perf

 Performance statistics for CORE#0
   Cycles       : 3523247483
   Instructions : 10093567
   CORE#0 performance (100MHz) : 0.29 MIPS
   Simulated time              : 35.23 s

 Performance statistics for CORE#1
   Cycles       : 3523247425
   Instructions : 20827648
   CORE#1 performance (100MHz) : 0.59 MIPS
   Simulated time              : 35.23 s

 Performance of laysim-gr712rc
   Real-time performance       : 12372.78 %
   laysim-gr712rc performance  : 108.59 MIPS
   Wall clock                  : 0.28 s
laysim>
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\RTEMS5.1-SMP> laysim-gr712rc-dbt-cli -core0 smp08.elf
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - smp08.elf
   - .text, addr: 0x40000000, size 143424 bytes
   - .rtemsroset, addr: 0x40023040, size 144 bytes
   - .rtemsstack, addr: 0x40023100/0x400230D0, size 32768 bytes (not loaded, just inf
o.)
   - .data, addr: 0x4002B100, size 2064 bytes
   - .bss, addr: 0x4002B940, size 28960 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
laysim> run

*** BEGIN OF TEST SMP 8 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
  CPU 1 running task Init - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA03 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA04 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA05 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA06 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:05   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA03 - rtems_clock_get_tod 09:00:15   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:15   12/31/1988
  CPU 0 running task TA04 - rtems_clock_get_tod 09:00:20   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:20   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:20   12/31/1988
  CPU 1 running task TA05 - rtems_clock_get_tod 09:00:25   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:25   12/31/1988
  CPU 0 running task TA03 - rtems_clock_get_tod 09:00:30   12/31/1988
  CPU 0 running task TA06 - rtems_clock_get_tod 09:00:30   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:30   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:30   12/31/1988

*** END OF TEST SMP 8 ***

CORE#1 halts because of Error Mode [0.422685sec/3523247483, 3523247425cycles]
laysim> perf

 Performance statistics for CORE#0
   Cycles       : 3523247483
   Instructions : 10093567
   CORE#0 performance (100MHz) : 0.29 MIPS
   Simulated time              : 35.23 s

 Performance statistics for CORE#1
   Cycles       : 3523247425
   Instructions : 20827648
   CORE#1 performance (100MHz) : 0.59 MIPS
   Simulated time              : 35.23 s

 Performance of laysim-gr712rc
   Real-time performance       : 8335.39 %
   laysim-gr712rc performance  : 73.15 MIPS
   Wall clock                  : 0.42 s
laysim>
```

3.2 Starting laysim-GR712RC

When launching `laysim-GR712RC`, many functions can be enabled through the start-up options described below.

- `-fast_uart` : `laysim-GR712RC` sets the baudrate of APBUART#0 as 38400bps for executing RAM executable. When '`-fast_uart`' option is used as start-up option, `laysim-GR712RC` runs APBUART#0 at infinite speed, rather than correct baudrate.

Figure 3-5. Using '-fast_uart' start-up option

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS RTEMS5.1-SMP]$ laysim-gr712rc-dbt-cli -r -core0 smp08.elf -fast_uart
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright


Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - smp08.elf
   - .text, addr: 0x40000000, size 143424 bytes
   - .rtemsroset, addr: 0x40023040, size 144 bytes
   - .rtemsstack, addr: 0x40023100/0x400230D0, size 32768 bytes (not loaded, just inf
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\RTEMS5.1-SMP> laysim-gr712rc-dbt-cli -r -core0 smp08.elf -fast_uar
t
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - smp08.elf
   - .text, addr: 0x40000000, size 143424 bytes
   - .rtemsroset, addr: 0x40023040, size 144 bytes
   - .rtemsstack, addr: 0x40023100/0x400230D0, size 32768 bytes (not loaded, just inf
```

```
o.)
   - .data, addr: 0x4002B100, size 2064 bytes
   - .bss, addr: 0x4002B940, size 28960 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

*** BEGIN OF TEST SMP 8 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
  CPU 1 running task Init - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA02 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA03 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA05 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA04 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA06 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:04   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:09   12/31/1988
  CPU 0 running task TA02 - rtems_clock_get_tod 09:00:09   12/31/1988
  CPU 1 running task TA03 - rtems_clock_get_tod 09:00:14   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:14   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:19   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:19   12/31/1988
  CPU 0 running task TA04 - rtems_clock_get_tod 09:00:19   12/31/1988
  CPU 1 running task TA05 - rtems_clock_get_tod 09:00:24   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:24   12/31/1988
  CPU 1 running task TA06 - rtems_clock_get_tod 09:00:29   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:29   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:29   12/31/1988
  CPU 0 running task TA03 - rtems_clock_get_tod 09:00:29   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:34   12/31/1988

*** END OF TEST SMP 8 ***


*** END OF TEST SMP 8 ***

CORE#1 halts because of Error Mode [0.291035sec/4017595284, 4017595229cycles]

 Performance statistics for CORE#0
   Cycles       : 4017595284
   Instructions : 11594205
   CORE#0 performance (100MHz) : 0.29 MIPS
   Simulated time              : 40.18 s

 Performance statistics for CORE#1
   Cycles       : 4017595229
   Instructions : 10070303
   CORE#1 performance (100MHz) : 0.25 MIPS
   Simulated time              : 40.18 s

 Performance of laysim-gr712rc
   Real-time performance       : 13803.42 %
   laysim-gr712rc performance  : 74.43 MIPS
   Wall clock                  : 0.29 s
[layright@CentOS RTEMS5.1-SMP]$
```

```
o.)
   - .data, addr: 0x4002B100, size 2064 bytes
   - .bss, addr: 0x4002B940, size 28960 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

*** BEGIN OF TEST SMP 8 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
  CPU 1 running task Init - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA02 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA03 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA05 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA04 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA06 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:04   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:09   12/31/1988
  CPU 0 running task TA02 - rtems_clock_get_tod 09:00:09   12/31/1988
  CPU 1 running task TA03 - rtems_clock_get_tod 09:00:14   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:14   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:19   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:19   12/31/1988
  CPU 0 running task TA04 - rtems_clock_get_tod 09:00:19   12/31/1988
  CPU 1 running task TA05 - rtems_clock_get_tod 09:00:24   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:24   12/31/1988
  CPU 1 running task TA06 - rtems_clock_get_tod 09:00:29   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:29   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:29   12/31/1988
  CPU 0 running task TA03 - rtems_clock_get_tod 09:00:29   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:34   12/31/1988

*** END OF TEST SMP 8 ***


*** END OF TEST SMP 8 ***

CORE#1 halts because of Error Mode [0.407837sec/4017595284, 4017595229cycles]

 Performance statistics for CORE#0
   Cycles       : 4017595284
   Instructions : 11594205
   CORE#0 performance (100MHz) : 0.29 MIPS
   Simulated time              : 40.18 s

 Performance statistics for CORE#1
   Cycles       : 4017595229
   Instructions : 10070303
   CORE#1 performance (100MHz) : 0.25 MIPS
   Simulated time              : 40.18 s

 Performance of laysim-gr712rc
   Real-time performance       : 9850.97 %
   laysim-gr712rc performance  : 53.12 MIPS
   Wall clock                  : 0.41 s
PS C:\laysim-gr712rc\RTEMS5.1-SMP>
```

■ `-core0/1` : Load an RAM executable for CORE#0/#1. This option is only available in the CLI version of `laysim-GR712RC`. It is needed to set the entry point and stack point in order to run AMP example in real hardware with GRMON3. `laysim-GR712RC` sets CORE#0/ CORE#1 `%sp/%fp` and `%pc` automatically according to loaded program address.

Figure 3-6. Run AMP example using '-core' start-up option

laysim-gr712rc-cli in Linux

```
[layright@CentOS BCC-2.2.1]$ laysim-gr712rc-cli -r -core0 bcc_amp0.elf -core1 bcc_amp1.
elf
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - bcc_amp0.elf
   - .text, addr: 0x40001000, size 58800 bytes
   - .rodata, addr: 0x4000F5B0, size 1904 bytes
   - .data, addr: 0x4000FD20, size 1560 bytes
   - .bss, addr: 0x40010338, size 424 bytes (not loaded, just info.)
CORE#1 RAM file is loaded - bcc_amp1.elf
   - .text, addr: 0x40100000, size 58704 bytes
   - .rodata, addr: 0x4010E550, size 1856 bytes
   - .data, addr: 0x4010EC90, size 1560 bytes
   - .bss, addr: 0x4010F2A8, size 424 bytes (not loaded, just info.)
Set CORE#0 %sp/%fp : 0x400FFFF0/0x40100000
Set CORE#1 %sp/%fp : 0x407FFFF0/0x40800000
GR712RC is configured as AMP, starts from RAM
amp0:main:25: ENTRY
amp0:main:27: bcc_get_cpu_count() => 2
amp0:main:28:
amp0:main:30: I am supposed to have cpuid=0
amp0:main:32: bcc_get_cpuid() => 0
amp0:main:39: I will now call bcc_start_processor(1)
amp0:main:40:
amp1:main:25: ENTRY
amp1:main:27: bcc_get_cpu_count() => 2
amp1:main:28:
amp1:main:30: I am supposed to have cpuid=1
amp1:main:32: bcc_get_cpuid() => 1
amp1:main:45: I am the node with the highest CPUID for this AMP application.
amp1:mai
CORE#1 halts because of Error Mode [13543262cycles/0.08202sec]
n:46:

 Performance statistics for CORE#0
   Cycles       : 13543287
   Instructions : 2214316
   Overall CPI  : 2.85
   CORE#0 performance (100MHz) : 35.12 MOPS (35.12 MIPS, 0.00 MFLOPS)
```

laysim-gr712rc-cli in Windows

```
PS C:\laysim-gr712rc\BCC-2.2.1> laysim-gr712rc-cli -r -core0 bcc_amp0.elf -core1 bcc_am
p1.elf
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - bcc_amp0.elf
   - .text, addr: 0x40001000, size 58800 bytes
   - .rodata, addr: 0x4000F5B0, size 1904 bytes
   - .data, addr: 0x4000FD20, size 1560 bytes
   - .bss, addr: 0x40010338, size 424 bytes (not loaded, just info.)
CORE#1 RAM file is loaded - bcc_amp1.elf
   - .text, addr: 0x40100000, size 58704 bytes
   - .rodata, addr: 0x4010E550, size 1856 bytes
   - .data, addr: 0x4010EC90, size 1560 bytes
   - .bss, addr: 0x4010F2A8, size 424 bytes (not loaded, just info.)
Set CORE#0 %sp/%fp : 0x400FFFF0/0x40100000
Set CORE#1 %sp/%fp : 0x407FFFF0/0x40800000
GR712RC is configured as AMP, starts from RAM
amp0:main:25: ENTRY
amp0:main:27: bcc_get_cpu_count() => 2
amp0:main:28:
amp0:main:30: I am supposed to have cpuid=0
amp0:main:32: bcc_get_cpuid() => 0
amp0:main:39: I will now call bcc_start_processor(1)
amp0:main:40:
amp1:main:25: ENTRY
amp1:main:27: bcc_get_cpu_count() => 2
amp1:main:28:
amp1:main:30: I am supposed to have cpuid=1
amp1:main:32: bcc_get_cpuid() => 1
amp1:main:45: I am the node with the highest CPUID for this AMP application.
amp1:mai
CORE#1 halts because of Error Mode [13543262cycles/0.121706sec]
n:46:

 Performance statistics for CORE#0
   Cycles       : 13543287
   Instructions : 2214316
   Overall CPI  : 2.85
   CORE#0 performance (100MHz) : 35.12 MOPS (35.12 MIPS, 0.00 MFLOPS)
```

```
   Simulated time              : 135.43 ms
   Processor utilization       : 46.55 %

 Performance statistics for CORE#1
   Cycles       : 13543262
   Instructions : 2540951
   Overall CPI  : 5.33
   CORE#1 performance (100MHz) : 18.76 MOPS (18.76 MIPS, 0.00 MFLOPS)
   Simulated time              : 135.43 ms
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 165.12 %
   laysim-gr712rc performance  : 57.98 MIPS
   Wall clock                  : 0.08 s
[layright@CentOS BCC-2.2.1]$
```

```
   Simulated time              : 135.43 ms
   Processor utilization       : 46.55 %

 Performance statistics for CORE#1
   Cycles       : 13543262
   Instructions : 2540951
   Overall CPI  : 5.33
   CORE#1 performance (100MHz) : 18.76 MOPS (18.76 MIPS, 0.00 MFLOPS)
   Simulated time              : 135.43 ms
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 111.28 %
   laysim-gr712rc performance  : 39.07 MIPS
   Wall clock                  : 0.12 s
PS C:\laysim-gr712rc\BCC-2.2.1>
```

■ `-rom` : Load a ROM executable to MRAM. This option is only available in the CLI version of `laysim-GR712RC`.

Figure 3-7. Run ROM example using '-rom' start-up option

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS BCC-2.2.1]$ laysim-gr712rc-dbt-cli -r -rom dhry-bin.rom
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
ROM file is loaded - dhry-bin.rom
   - binary file size : 43872 bytes
GR712RC ROM file is loaded, starts from ROM


  MKPROM2 boot loader v2.0.65
  Copyright Cobham Gaisler AB - all rights reserved

  system clock    : 100.0 MHz
  baud rate       : 38343 baud
  prom            : 8192 K, (15/15) ws (r/w)
  sram            : 8192 K, 1 bank(s), 2/2 ws (r/w)

  decompressing .text to 0x40000000
  decompressing .rodata to 0x4000eb00
  decompressing .data to 0x4000f300

  starting dhry.elf

Execution starts, 1000000 runs through Dhrystone
Total execution time:                      5.8 s
Microseconds for one run through Dhrystone:    5.8
Dhrystones per Second:                  172711.6

Dhrystones MIPS      :                    98.3

CORE#0 halts because of Error Mode [0.505957sec/598916412, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 598916412
   Instructions : 367217418
   CORE#0 performance (100MHz) : 61.31 MIPS
   Simulated time              : 5.99 s

 Performance of laysim-gr712rc
   Real-time performance       : 1183.73 %
   laysim-gr712rc performance  : 725.79 MIPS
   Wall clock                  : 0.51 s
[layright@CentOS BCC-2.2.1]$
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\BCC-2.2.1> laysim-gr712rc-dbt-cli -r -rom dhry-bin.rom
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
ROM file is loaded - dhry-bin.rom
   - binary file size : 43872 bytes
GR712RC ROM file is loaded, starts from ROM


  MKPROM2 boot loader v2.0.65
  Copyright Cobham Gaisler AB - all rights reserved

  system clock    : 100.0 MHz
  baud rate       : 38343 baud
  prom            : 8192 K, (15/15) ws (r/w)
  sram            : 8192 K, 1 bank(s), 2/2 ws (r/w)

  decompressing .text to 0x40000000
  decompressing .rodata to 0x4000eb00
  decompressing .data to 0x4000f300

  starting dhry.elf

Execution starts, 1000000 runs through Dhrystone
Total execution time:                      5.8 s
Microseconds for one run through Dhrystone:    5.8
Dhrystones per Second:                  172711.6

Dhrystones MIPS      :                    98.3

CORE#0 halts because of Error Mode [0.46883sec/598916412, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 598916412
   Instructions : 367217418
   CORE#0 performance (100MHz) : 61.31 MIPS
   Simulated time              : 5.99 s

 Performance of laysim-gr712rc
   Real-time performance       : 1277.47 %
   laysim-gr712rc performance  : 783.26 MIPS
   Wall clock                  : 0.47 s
PS C:\laysim-gr712rc\BCC-2.2.1>
```

■ `-r` : Set auto-start & auto-exit with reporting the performance statistics. This option is only available in the CLI version of `laysim-GR712RC`.

Figure 3-8. Using '-r' start-up option

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS RTEMS5.1-SMP]$ laysim-gr712rc-dbt-cli -r -core0 smp09.elf
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - smp09.elf
   - .text, addr: 0x40000000, size 118192 bytes
   - .rtemsroset, addr: 0x4001CDB0, size 144 bytes
   - .rtemsstack, addr: 0x4001CE40, size 32768 bytes (not loaded, just info.)
   - .data, addr: 0x40024E40, size 1328 bytes
   - .bss, addr: 0x40025380, size 17952 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000


*** BEGIN OF TEST SMP 9 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
 CPU 1 start task TA1
 kill 10 clock ticks
--------------------------------------------------------------------------------
                                CPU USAGE BY THREAD
------------+-----------------------------------+--------------+---------
 ID         | NAME                              | SECONDS      | PERCENT
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\RTEMS5.1-SMP> laysim-gr712rc-dbt-cli -r -core0 smp09.elf
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - smp09.elf
   - .text, addr: 0x40000000, size 118192 bytes
   - .rtemsroset, addr: 0x4001CDB0, size 144 bytes
   - .rtemsstack, addr: 0x4001CE40, size 32768 bytes (not loaded, just info.)
   - .data, addr: 0x40024E40, size 1328 bytes
   - .bss, addr: 0x40025380, size 17952 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000


*** BEGIN OF TEST SMP 9 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
 CPU 1 start task TA1
 kill 10 clock ticks
--------------------------------------------------------------------------------
                                CPU USAGE BY THREAD
------------+-----------------------------------+--------------+---------
 ID         | NAME                              | SECONDS      | PERCENT
```

```
-------------+-----------------------------------------+---------------+---------
 0x09010001 | IDLE                                    |      1.200689 |  91.229
 0x09010002 | IDLE                                    |      1.000049 |  74.665
 0x0a010001 | UI1                                     |      0.362585 |  26.609
 0x0a010002 | TA1                                     |      0.185203 |  13.363
-------------+-----------------------------------------+---------------+---------
 TIME SINCE LAST CPU USAGE RESET IN SECONDS:                    1.385895
------------------------------------------------------------------------------

*** END OF TEST SMP 9 ***

CORE#1 halts because of Error Mode [0.056623sec/66502121, 66501899cycles]

 Performance statistics for CORE#0
   Cycles       : 66502121
   Instructions : 226859
   CORE#0 performance (100MHz) : 0.34 MIPS
   Simulated time              : 665.02 ms

 Performance statistics for CORE#1
   Cycles       : 66501899
   Instructions : 29067376
   CORE#1 performance (100MHz) : 43.71 MIPS
   Simulated time              : 665.02 ms

 Performance of laysim-gr712rc
   Real-time performance       : 1173.93 %
   laysim-gr712rc performance  : 517.12 MIPS
   Wall clock                  : 0.06 s
[layright@CentOS RTEMS5.1-SMP]$
```

```
-------------+-----------------------------------------+---------------+---------
 0x09010001 | IDLE                                    |      1.200689 |  91.229
 0x09010002 | IDLE                                    |      1.000049 |  74.665
 0x0a010001 | UI1                                     |      0.362585 |  26.609
 0x0a010002 | TA1                                     |      0.185203 |  13.363
-------------+-----------------------------------------+---------------+---------
 TIME SINCE LAST CPU USAGE RESET IN SECONDS:                    1.385895
------------------------------------------------------------------------------

*** END OF TEST SMP 9 ***

CORE#1 halts because of Error Mode [0.15552sec/66502121, 66501899cycles]

 Performance statistics for CORE#0
   Cycles       : 66502121
   Instructions : 226859
   CORE#0 performance (100MHz) : 0.34 MIPS
   Simulated time              : 665.02 ms

 Performance statistics for CORE#1
   Cycles       : 66501899
   Instructions : 29067376
   CORE#1 performance (100MHz) : 43.71 MIPS
   Simulated time              : 665.02 ms

 Performance of laysim-gr712rc
   Real-time performance       : 376.58 %
   laysim-gr712rc performance  : 165.88 MIPS
   Wall clock                  : 0.18 s
PS C:\laysim-gr712rc\RTEMS5.1-SMP>
```

■ `-traplog` : Log a synchronous trap information on occurred except Window underflow/overflow and interrupts.

Figure 3-9. Using '-traplog' start-up option

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS RCC-1.3.1]$ laysim-gr712rc-dbt-cli -r -core0 rtems-synctrap.elf -trapl
og
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - rtems-synctrap.elf
   - .text, addr: 0x40000000, size 160800 bytes
   - .rtemsroset, addr: 0x40027420, size 160 bytes
   - .rtemsstack, addr: 0x400274C0, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x400294C0, size 4720 bytes
   - .bss, addr: 0x4002A740, size 14384 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

Unaligned Memory (tt=0x07) occurred at 545263 cycles

 ---------------------- CORE#0 ----------------------
        INS       LOCALS       OUTS       GLOBALS
  0:  4002D4E0    F3000FC2    00000000    00000000
  1:  4002EE98    40013DFC    00000190    00000001
  2:  4002A204    40013E00    4002EE4C    00000000
  3:  00000001    F30000C2    00000000    00200000
  4:  400291D0    00000000    40026800    00000000
  5:  40001000    00000020    4002BD10    00000000
  6:  4002EE50    FFFFFFFE    4002EDE8    4002CF40
  7:  40010858    FFFFFFFF    40001314    00000000

 psr:F34000E2 wim:00000008 tbr:400008A0 y:FFFFFFFE
 [n:0 z:1 v:0 c:0 ef:0 pil:0 s:1 ps:1 et:1 cwp:2]

 pc:    4000134C : D0004000   ld      [%g1], %o0
 npc:   40001350 : 91D02010   ta      0x10
Caught synchronous trap with vector 0x07
Caught synchronous trap with vector 0x90
CORE#0 halts because of Error Mode [0.006602sec/2993998, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 2993998
   Instructions : 1256906
   CORE#0 performance (100MHz) : 41.98 MIPS
   Simulated time              : 29.94 ms

 Performance of laysim-gr712rc
   Real-time performance       : 453.50 %
   laysim-gr712rc performance  : 190.38 MIPS
   Wall clock                  : 0.01 s
[layright@CentOS RCC-1.3.1]$
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\RCC-1.3.1> laysim-gr712rc-dbt-cli -r -core0 rtems-synctrap.elf -tr
aplog
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - rtems-synctrap.elf
   - .text, addr: 0x40000000, size 160800 bytes
   - .rtemsroset, addr: 0x40027420, size 160 bytes
   - .rtemsstack, addr: 0x400274C0, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x400294C0, size 4720 bytes
   - .bss, addr: 0x4002A740, size 14384 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

Unaligned Memory (tt=0x07) occurred at 545263 cycles

 ---------------------- CORE#0 ----------------------
        INS       LOCALS       OUTS       GLOBALS
  0:  4002D4E0    F3000FC2    00000000    00000000
  1:  4002EE98    40013DFC    00000190    00000001
  2:  4002A204    40013E00    4002EE4C    00000000
  3:  00000001    F30000C2    00000000    00200000
  4:  400291D0    00000000    40026800    00000000
  5:  40001000    00000020    4002BD10    00000000
  6:  4002EE50    FFFFFFFE    4002EDE8    4002CF40
  7:  40010858    FFFFFFFF    40001314    00000000

 psr:F34000E2 wim:00000008 tbr:400008A0 y:FFFFFFFE
 [n:0 z:1 v:0 c:0 ef:0 pil:0 s:1 ps:1 et:1 cwp:2]

 pc:    4000134C : D0004000   ld      [%g1], %o0
 npc:   40001350 : 91D02010   ta      0x10
Caught synchronous trap with vector 0x07
Caught synchronous trap with vector 0x90
CORE#0 halts because of Error Mode [0.0298633sec/2993998, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 2993998
   Instructions : 1256906
   CORE#0 performance (100MHz) : 41.98 MIPS
   Simulated time              : 29.94 ms

 Performance of laysim-gr712rc
   Real-time performance       : 100.26 %
   laysim-gr712rc performance  : 42.09 MIPS
   Wall clock                  : 0.03 s
PS C:\laysim-gr712rc\RCC-1.3.1>
```

■ `-batch` : Run commands in a batch file. This option is only available in the CLI version of `laysim-GR712RC`.

Figure 3-10. Using '-batch' start-up option

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS RCC-1.3.1]$ cat gr712rc.bat
load rtems-hello.elf
b Init
t 10
run
reg
perf
quit

[layright@CentOS RCC-1.3.1]$ laysim-gr712rc-dbt-cli -batch gr712rc.bat
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\RCC-1.3.1> cat gr712rc.bat
load rtems-hello.elf
b Init
t 10
run
reg
perf
quit

PS C:\laysim-gr712rc\RCC-1.3.1> laysim-gr712rc-dbt-cli -batch gr712rc.bat
```

```
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
--> No CORE#0 RAM file is loaded
Read batch file from gr712rc.bat
CORE#0 RAM file is loaded - rtems-hello.elf
   - .text, addr: 0x40000000, size 170800 bytes
   - .rtemsroset, addr: 0x40029B30, size 160 bytes
   - .rtemsstack, addr: 0x40029C00, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x4002BC00, size 4720 bytes
   - .bss, addr: 0x4002CE80, size 15792 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

[CORE#0] Set breakpoint at Init(0x400012CC)

       0 (0) E CORE#0   40000000 : A0100000   or      %g0, %g0, %10   ! start
       1 (1) E CORE#0   40000004 : 29100004   sethi   %hi(0x40001000), %l4
       2 (2) E CORE#0   40000008 : 81C52000   jmpl    %l4, %g0
       4 (4) E CORE#0   4000000C : A6102000   or      %g0, 0x0, %l3
       5 (5) E CORE#0   40001000 : 03100000   sethi   %hi(0x40000000), %g1   ! hard_res
et
       6 (6) E CORE#0   40001004 : 82106000   or      %g1, 0x0, %g1
       7 (7) E CORE#0   40001008 : 81980001   wr      %g0, %g1, %tbr
       8 (8) E CORE#0   4000100C : 83480000   rd      %psr, %g1
       9 (9) E CORE#0   40001010 : 84006001   add     %g1, 0x1, %g2
      10 (10) E CORE#0   40001014 : 8408A007   and     %g2, 0x7, %g2

[CORE#0] Breakpoint hit at 0x400012CC(Init)


 ---------------------- CORE#0 ----------------------
        INS       LOCALS       OUTS       GLOBALS
  0:  4002D040    40031B58    400301A0    00000000
  1:  40013B18    40027ACC    40031B58    400012CC
  2:  00000000    4002C958    4002C944    0A010001
  3:  4002C944    4002C800    00000001    0A010001
  4:  4002D040    4002C95C    4002B910    40031B5C
  5:  40027B14    4002C920    400278F0    00000000
  6:  40031B70    4002D310    40031B10    4002FC00
  7:  4000F2D0    4002D09C    40010778    00000000

 psr:F30000E3 wim:00000040 tbr:400008A0 y:FFFFFFFE
 [n:0 z:0 v:0 c:0 ef:0 pil:0 s:1 ps:1 et:1 cwp:3]

 pc:    400012CC : 9DE3BFA0   save    %sp, -96, %sp
 npc:   400012D0 : 1110009E   sethi   %hi(0x40027800), %o0

 Performance statistics for CORE#0
   Cycles       : 545727
   Instructions : 345493
   CORE#0 performance (100MHz) : 63.31 MIPS
   Simulated time              : 5.46 ms

 Performance of laysim-gr712rc
   Real-time performance       : 62.77 %
   laysim-gr712rc performance  : 39.74 MIPS
   Wall clock                  : 0.01 s
[layright@CentOS RCC-1.3.1]$
```

```
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
--> No CORE#0 RAM file is loaded
Read batch file from gr712rc.bat
CORE#0 RAM file is loaded - rtems-hello.elf
   - .text, addr: 0x40000000, size 170800 bytes
   - .rtemsroset, addr: 0x40029B30, size 160 bytes
   - .rtemsstack, addr: 0x40029C00, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x4002BC00, size 4720 bytes
   - .bss, addr: 0x4002CE80, size 15792 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

[CORE#0] Set breakpoint at Init(0x400012CC)

       0 (0) E CORE#0   40000000 : A0100000   or      %g0, %g0, %10   ! start
       1 (1) E CORE#0   40000004 : 29100004   sethi   %hi(0x40001000), %l4
       2 (2) E CORE#0   40000008 : 81C52000   jmpl    %l4, %g0
       4 (4) E CORE#0   4000000C : A6102000   or      %g0, 0x0, %l3
       5 (5) E CORE#0   40001000 : 03100000   sethi   %hi(0x40000000), %g1   ! hard_res
et
       6 (6) E CORE#0   40001004 : 82106000   or      %g1, 0x0, %g1
       7 (7) E CORE#0   40001008 : 81980001   wr      %g0, %g1, %tbr
       8 (8) E CORE#0   4000100C : 83480000   rd      %psr, %g1
       9 (9) E CORE#0   40001010 : 84006001   add     %g1, 0x1, %g2
      10 (10) E CORE#0   40001014 : 8408A007   and     %g2, 0x7, %g2

[CORE#0] Breakpoint hit at 0x400012CC(Init)


 ---------------------- CORE#0 ----------------------
        INS       LOCALS       OUTS       GLOBALS
  0:  4002D040    40031B58    400301A0    00000000
  1:  40013B18    40027ACC    40031B58    400012CC
  2:  00000000    4002C958    4002C944    0A010001
  3:  4002C944    4002C800    00000001    0A010001
  4:  4002D040    4002C95C    4002B910    40031B5C
  5:  40027B14    4002C920    400278F0    00000000
  6:  40031B70    4002D310    40031B10    4002FC00
  7:  4000F2D0    4002D09C    40010778    00000000

 psr:F30000E3 wim:00000040 tbr:400008A0 y:FFFFFFFE
 [n:0 z:0 v:0 c:0 ef:0 pil:0 s:1 ps:1 et:1 cwp:3]

 pc:    400012CC : 9DE3BFA0   save    %sp, -96, %sp
 npc:   400012D0 : 1110009E   sethi   %hi(0x40027800), %o0

 Performance statistics for CORE#0
   Cycles       : 545727
   Instructions : 345493
   CORE#0 performance (100MHz) : 63.31 MIPS
   Simulated time              : 5.46 ms

 Performance of laysim-gr712rc
   Real-time performance       : 18.87 %
   laysim-gr712rc performance  : 11.95 MIPS
   Wall clock                  : 0.03 s
PS C:\laysim-gr712rc\RCC-1.3.1>
```

■ `-cov` : Enable code coverage. `laysim-gr712rc-cli` only supports for the code coverage in the same format as `TSIM3-LEON3`. '`-cov`' start-up option should be enabled for the code coverage. Detailed information can be found in `[laysim-GR712RC-004] Code coverage operation on laysim-GR712RC`.

Figure 3-11. Using '-cov' start-up option

laysim-gr712rc-cli in Linux

```
[layright@CentOS RTEMS5.1-SMP]$ laysim-gr712rc-cli -r -core0 smp09.elf -cov
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - smp09.elf
   - .text, addr: 0x40000000, size 118192 bytes
   - .rtemsroset, addr: 0x4001CDB0, size 144 bytes
   - .rtemsstack, addr: 0x4001CE40, size 32768 bytes (not loaded, just info.)
   - .data, addr: 0x40024E40, size 1328 bytes
   - .bss, addr: 0x40025380, size 17952 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000


*** BEGIN OF TEST SMP 9 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
 CPU 1 start task TA1
 kill 10 clock ticks
--------------------------------------------------------------------------------
                                CPU USAGE BY THREAD
------------+-----------------------------------------+--------------+---------
 ID         | NAME                                    | SECONDS      | PERCENT
------------+-----------------------------------------+--------------+---------
 0x09010001 | IDLE                                    |     1.171035 |  91.028
 0x09010002 | IDLE                                    |     1.000131 |  76.363
 0x0a010001 | UI1                                     |     0.332825 |  24.968
 0x0a010002 | TA1                                     |     0.185180 |  13.654
------------+-----------------------------------------+--------------+---------
 TIME SINCE LAST CPU USAGE RESET IN SECONDS:                          1.356217
--------------------------------------------------------------------------------

*** END OF TEST SMP 9 ***


CORE#1 halts because of Error Mode [64222487cycles/0.558125sec]
```

laysim-gr712rc-cli in Windows

```
PS C:\laysim-gr712rc\RTEMS5.1-SMP> laysim-gr712rc-cli -r -core0 smp09.elf -cov
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - smp09.elf
   - .text, addr: 0x40000000, size 118192 bytes
   - .rtemsroset, addr: 0x4001CDB0, size 144 bytes
   - .rtemsstack, addr: 0x4001CE40, size 32768 bytes (not loaded, just info.)
   - .data, addr: 0x40024E40, size 1328 bytes
   - .bss, addr: 0x40025380, size 17952 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000


*** BEGIN OF TEST SMP 9 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
 CPU 1 start task TA1
 kill 10 clock ticks
--------------------------------------------------------------------------------
                                CPU USAGE BY THREAD
------------+-----------------------------------------+--------------+---------
 ID         | NAME                                    | SECONDS      | PERCENT
------------+-----------------------------------------+--------------+---------
 0x09010001 | IDLE                                    |     1.171035 |  91.028
 0x09010002 | IDLE                                    |     1.000131 |  76.363
 0x0a010001 | UI1                                     |     0.332825 |  24.968
 0x0a010002 | TA1                                     |     0.185180 |  13.654
------------+-----------------------------------------+--------------+---------
 TIME SINCE LAST CPU USAGE RESET IN SECONDS:                          1.356217
--------------------------------------------------------------------------------

*** END OF TEST SMP 9 ***


CORE#1 halts because of Error Mode [64222487cycles/0.68989sec]
```

```
 Performance statistics for CORE#0
   Cycles       : 64222527
   Instructions : 215588
   Overall CPI  : 5.09
   CORE#0 performance (100MHz) : 19.65 MOPS (19.65 MIPS, 0.00 MFLOPS)
   Simulated time              : 642.23 ms
   Processor utilization       : 1.71 %

 Performance statistics for CORE#1
   Cycles       : 64222487
   Instructions : 32157740
   Overall CPI  : 2.00
   CORE#1 performance (100MHz) : 50.07 MOPS (50.07 MIPS, 0.00 MFLOPS)
   Simulated time              : 642.22 ms
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 115.07 %
   laysim-gr712rc performance  : 58.00 MIPS
   Wall clock                  : 0.56 s

[layright@CentOS RTEMS5.1-SMP]$ ls -als smp09.elf.cov
56 -rw-rw-r--. 1 layright layright 53552 Oct 23 17:38 smp09.elf.cov
[layright@CentOS RTEMS5.1-SMP]$
```

```
 Performance statistics for CORE#0
   Cycles       : 64222527
   Instructions : 215588
   Overall CPI  : 5.09
   CORE#0 performance (100MHz) : 19.65 MOPS (19.65 MIPS, 0.00 MFLOPS)
   Simulated time              : 642.23 ms
   Processor utilization       : 1.71 %

 Performance statistics for CORE#1
   Cycles       : 64222487
   Instructions : 32157740
   Overall CPI  : 2.00
   CORE#1 performance (100MHz) : 50.07 MOPS (50.07 MIPS, 0.00 MFLOPS)
   Simulated time              : 642.22 ms
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 93.09 %
   laysim-gr712rc performance  : 46.93 MIPS
   Wall clock                  : 0.69 s

PS C:\laysim-gr712rc\RTEMS5.1-SMP> ls smp09.elf.cov

    Directory: C:\laysim-gr712rc\RTEMS5.1-SMP

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----       10/23/2025   5:39 PM          54246 smp09.elf.cov

PS C:\laysim-gr712rc\RTEMS5.1-SMP>
```

```
40000000 : 7 3 3 3 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 1 19 19 0 1 1 1 1 6 2 2 2
40000080 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000100 : 6 6 6 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000180 : 7 3 3 3 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 7 3 3 3 6 2 2 2
40000200 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000280 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000300 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000380 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000400 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000480 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000500 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000580 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000600 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000680 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000780 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
40000800 : 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000880 : 6 2 2 2 19 19 19 19 19 19 19 19 9 1 9 19 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000900 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000980 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000a00 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000a80 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000b00 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000b80 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
...
...
4002ec80 : 2 2 2 2 2 2 0 6 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2
4002ed00 : 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2
4002ed80 : 2 2 0 6 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2 2 2 2 2 2
4002ee00 : 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6
4002ee80 : 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2 6 2 2 2 2 2 2 2 2 2
4002ef00 : 2 2 2 2 2 2 2 0 6 6 6 6 2 6 2 6 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2
4002ef80 : 6 2 2 2 0 6 6 6 2 2 6 2 6 2 0 6 6 6 2 2 6 2 6 2 0 6 6 6 2 2 6 6
4002f000 : 6 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2
4002f080 : 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6
4002f100 : 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6
4002f180 : 2 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2
4002f200 : 6 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2 6 2 6 2 2 2 2 2 2 2 2 2 2 2 2
4002f280 : 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2
4002f300 : 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6
4002f380 : 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6
4002f400 : 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2
4002f480 : 2 2 2 2 0 6 6 6 2 2 6 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2 6 2
4002f500 : 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2 6 2 2 2 2 2 2 2 2 2 2 2 2 2
4002f580 : 0 6 6 6 2 2 6 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2 6 2 2 2 2 2 2
4002f600 : 2 2 2 2 2 2 2 2 0 6 6 6 6 6 2 2 2 6 2 6 6 2 2 2 2 2 2 2 2 6 6 6
4002f680 : 6 2 2 2 0 6 6 6 2 6 6 6 6 6 6 6 6 6 6 6 2 6 2 6 6 6 0 0 2 2 2
407fff80 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 6 0 0 0 0
```

```
40000000 : 7 3 3 3 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 1 19 19 0 1 1 1 1 6 2 2 2
40000080 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000100 : 6 6 6 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000180 : 7 3 3 3 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 7 3 3 3 6 2 2 2
40000200 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000280 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000300 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000380 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000400 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000480 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000500 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000580 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000600 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000680 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000780 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
40000800 : 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000880 : 6 2 2 2 19 19 19 19 19 19 19 19 9 1 9 19 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000900 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000980 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000a00 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000a80 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000b00 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000b80 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
...
...
4002ec80 : 2 2 2 2 2 2 0 6 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2
4002ed00 : 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2
4002ed80 : 2 2 0 6 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2 2 2 2 2 2
4002ee00 : 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6
4002ee80 : 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2 6 2 2 2 2 2 2 2 2 2
4002ef00 : 2 2 2 2 2 2 2 0 6 6 6 6 2 6 2 6 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2
4002ef80 : 6 2 2 2 0 6 6 6 2 2 6 2 6 2 0 6 6 6 2 2 6 2 6 2 0 6 6 6 2 2 6 6
4002f000 : 6 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2
4002f080 : 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6
4002f100 : 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6
4002f180 : 2 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2
4002f200 : 6 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2 6 2 6 2 2 2 2 2 2 2 2 2 2 2 2
4002f280 : 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2
4002f300 : 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6
4002f380 : 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6
4002f400 : 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2
4002f480 : 2 2 2 2 0 6 6 6 2 2 6 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2 6 2
4002f500 : 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2 6 2 2 2 2 2 2 2 2 2 2 2 2 2
4002f580 : 0 6 6 6 2 2 6 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2 6 2 2 2 2 2 2
4002f600 : 2 2 2 2 2 2 2 2 0 6 6 6 6 6 2 2 2 6 2 6 6 2 2 2 2 2 2 2 2 6 6 6
4002f680 : 6 2 2 2 0 6 6 6 2 6 6 6 6 6 6 6 6 6 6 6 2 6 2 6 6 6 0 0 2 2 2
407fff80 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 6 0 0 0 0
```

■ `-nb` : Runs the other core even if one of the two cores enters error mode. By default, if any core enters error mode, `laysim-GR712RC` halts execution. However, when the `-nb` option is enabled, execution continues on the other core even if one core transitions to error mode. `laysim-GR712RC` only halts execution when both cores enter error mode.

Figure 3-12. Using '-nb' start-up option

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS RTEMS5.1-SMP]$ laysim-gr712rc-dbt-cli -r -core0 smp09.elf -nb
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - smp09.elf
   - .text, addr: 0x40000000, size 118192 bytes
   - .rtemsroset, addr: 0x4001CDB0, size 144 bytes
   - .rtemsstack, addr: 0x4001CE40, size 32768 bytes (not loaded, just info.)
   - .data, addr: 0x40024E40, size 1328 bytes
   - .bss, addr: 0x40025380, size 17952 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000


*** BEGIN OF TEST SMP 9 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\RTEMS5.1-SMP> laysim-gr712rc-dbt-cli -r -core0 smp09.elf -nb
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - smp09.elf
   - .text, addr: 0x40000000, size 118192 bytes
   - .rtemsroset, addr: 0x4001CDB0, size 144 bytes
   - .rtemsstack, addr: 0x4001CE40, size 32768 bytes (not loaded, just info.)
   - .data, addr: 0x40024E40, size 1328 bytes
   - .bss, addr: 0x40025380, size 17952 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000


*** BEGIN OF TEST SMP 9 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
```

```
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
 CPU 1 start task TA1
 kill 10 clock ticks
-------------------------------------------------------------------------------
                                CPU USAGE BY THREAD
------------+-----------------------------------------+---------------+---------
 ID         | NAME                                    | SECONDS       | PERCENT
------------+-----------------------------------------+---------------+---------
 0x09010001 | IDLE                                    |      1.200689 |  91.229
 0x09010002 | IDLE                                    |      1.000049 |  74.665
 0x0a010001 | UI1                                     |      0.362585 |  26.609
 0x0a010002 | TA1                                     |      0.185203 |  13.363
------------+-----------------------------------------+---------------+---------
 TIME SINCE LAST CPU USAGE RESET IN SECONDS:                          1.385895
-------------------------------------------------------------------------------

*** END OF TEST SMP 9 ***

CORE#1 halts because of Error Mode [0.053822sec/66502121, 66501899cycles]

laysim> core
GR712RC CORE#0 is execution mode
GR712RC CORE#1 is error mode
GR712RC CORE#0 is selected for debugging

laysim> step
72367555297 (72367555133) E CORE#0   40001B7C : C0838020   lda     [%sp] 0x1, %g0

laysim> step
72367555297 (72367555136) E CORE#0   40001B80 : 30BFFFFE   ba,a    0x40001b78

laysim> step
72367555297 (72367555138) E CORE#0   40001B78 : A7800000   wr      %g0, %g0, %asr19   !
leon3_power_down_loop

laysim>
```

```
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
 CPU 1 start task TA1
 kill 10 clock ticks
-------------------------------------------------------------------------------
                                CPU USAGE BY THREAD
------------+-----------------------------------------+---------------+---------
 ID         | NAME                                    | SECONDS       | PERCENT
------------+-----------------------------------------+---------------+---------
 0x09010001 | IDLE                                    |      1.200689 |  91.229
 0x09010002 | IDLE                                    |      1.000049 |  74.665
 0x0a010001 | UI1                                     |      0.362585 |  26.609
 0x0a010002 | TA1                                     |      0.185203 |  13.363
------------+-----------------------------------------+---------------+---------
 TIME SINCE LAST CPU USAGE RESET IN SECONDS:                          1.385895
-------------------------------------------------------------------------------

*** END OF TEST SMP 9 ***

CORE#1 halts because of Error Mode [0.1731sec/66502121, 66501899cycles]

laysim> core
GR712RC CORE#0 is execution mode
GR712RC CORE#1 is error mode
GR712RC CORE#0 is selected for debugging

laysim> step
75226719497 (75226719333) E CORE#0   40001B7C : C0838020   lda     [%sp] 0x1, %g0

laysim> step
75226719497 (75226719336) E CORE#0   40001B80 : 30BFFFFE   ba,a    0x40001b78

laysim> step
75226719497 (75226719338) E CORE#0   40001B78 : A7800000   wr      %g0, %g0, %asr19   !
leon3_power_down_loop

laysim>
```

■ `-ramsz` : `laysim-GR712RC` provides 8MB of SRAM by default, and the SRAM size can be changed using the `-ramsz` option. The updated SRAM size is reflected in `FTMCTRL.MCFG2.SZ`, and if RAM code is loaded `%sp`/`%fp` are updated according to the new SRAM size.

• 8 : 8KB, 16 : 16KB, 32 : 32KB, 64 : 64KB, 128 : 128KB, 256 : 256KB, 512 : 512KB, 1024 : 1MB, 2048 : 2MB, 4096 : 4MB,

• 8192 : 8MB (default), 16384 : 16MB, 32768 : 32MB (GR712RC Max SRAM size), 65536 : 64MB, 131072 : 128MB, 262144 : 256MB

Figure 3-13. Using '-ramsz' start-up option

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS Linux]$ laysim-gr712rc-dbt-cli -r -core0 gr712rc-linux-5.10-1.9-SMP.el
f -ramsz 32768
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - gr712rc-linux-5.10-1.9-SMP.elf
   - .text, addr: 0x40000000, size 3904 bytes
   - .data, addr: 0x40000F40, size 80 bytes
   - .bss, addr: 0x40000F90, size 16 bytes (not loaded, just info.)
   - .mmu_tables, addr: 0x40002000/0x40000F90, size 2560 bytes (not loaded, just info.)
   - .vmlinux, addr: 0x40004000, size 8751252 bytes
   - .startup_bss, addr: 0xFFD03000, size 640 bytes (not loaded, just info.)
   - .ps_move_startup, addr: 0xFFD03280/0x4085C8A0, size 36096 bytes
   - .ps_move_prom, addr: 0xFFD0BF80/0x408655A0, size 5136 bytes
   - .prom_bss, addr: 0xFFD0D390, size 80 bytes (not loaded, just info.)
   - .ps_stack, addr: 0xFFD133E0, size 3104 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x41FFFFF0/0x42000000

PROMLIB: Sun Boot Prom Version 0 Revision 0
Linux version 5.10.237-leon1.9 (ludwig@ludwig.got.gaisler.com) (sparc-gaisler-linux5.10
-gcc (LEON Linux 5.10 Toolchain 2.1) 13.2.1 20240119, GNU ld (GNU Binutils) 2.42) #4 SM
P Fri Jun 27 13:57:42 CEST 2025
printk: bootconsole [earlyprom0] enabled
ARCH: LEON
TYPE: LEON3 System-on-a-Chip
Ethernet address: 00:00:7c:cc:01:45
CACHE: 4-way associative cache, set size 4k
Zone ranges:
  DMA      [mem 0x0000000040000000-0x0000000041ff6fff]
  Normal   empty
  HighMem  empty
Movable zone start for each node
Early memory node ranges
  node   0: [mem 0x0000000040000000-0x0000000041ff6fff]
Initmem setup node 0 [mem 0x0000000040000000-0x0000000041ff6fff]
OF stdout device is: /a::a
PROM: Built device tree with 52609 bytes of memory.
Booting Linux...
percpu: Embedded 10 pages/cpu s17708 r0 d23252 u40960
Built 1 zonelists, mobility grouping off.  Total pages: 8119
Kernel command line: console=ttyS0,38400
Dentry cache hash table entries: 4096 (order: 2, 16384 bytes, linear)
Inode-cache hash table entries: 2048 (order: 1, 8192 bytes, linear)
Sorting __ex_table...
mem auto-init: stack:all(zero), heap alloc:off, heap free:off
Memory: 21228K/32732K available (4297K kernel code, 202K rwdata, 648K rodata, 3388K ini
t, 274K bss, 11504K reserved, 0K cma-reserved, 0K highmem)
rcu: Hierarchical RCU implementation.
rcu:          RCU event tracing is enabled.
rcu:          RCU restricting CPUs from NR_CPUS=32 to nr_cpu_ids=2.
rcu: RCU calculated value of scheduler-enlistment delay is 10 jiffies.
rcu: Adjusting geometry for rcu_fanout_leaf=16, nr_cpu_ids=2
NR_IRQS: 64
Console: colour dummy device 80x25
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\Linux> laysim-gr712rc-dbt-cli -r -core0 gr712rc-linux-5.10-1.9-SM
P.elf -ramsz 32768
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - gr712rc-linux-5.10-1.9-SMP.elf
   - .text, addr: 0x40000000, size 3904 bytes
   - .data, addr: 0x40000F40, size 80 bytes
   - .bss, addr: 0x40000F90, size 16 bytes (not loaded, just info.)
   - .mmu_tables, addr: 0x40002000/0x40000F90, size 2560 bytes (not loaded, just info.)
   - .vmlinux, addr: 0x40004000, size 8751252 bytes
   - .startup_bss, addr: 0xFFD03000, size 640 bytes (not loaded, just info.)
   - .ps_move_startup, addr: 0xFFD03280/0x4085C8A0, size 36096 bytes
   - .ps_move_prom, addr: 0xFFD0BF80/0x408655A0, size 5136 bytes
   - .prom_bss, addr: 0xFFD0D390, size 80 bytes (not loaded, just info.)
   - .ps_stack, addr: 0xFFD133E0, size 3104 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x41FFFFF0/0x42000000

PROMLIB: Sun Boot Prom Version 0 Revision 0
Linux version 5.10.237-leon1.9 (ludwig@ludwig.got.gaisler.com) (sparc-gaisler-linux5.10
-gcc (LEON Linux 5.10 Toolchain 2.1) 13.2.1 20240119, GNU ld (GNU Binutils) 2.42) #4 SM
P Fri Jun 27 13:57:42 CEST 2025
printk: bootconsole [earlyprom0] enabled
ARCH: LEON
TYPE: LEON3 System-on-a-Chip
Ethernet address: 00:00:7c:cc:01:45
CACHE: 4-way associative cache, set size 4k
Zone ranges:
  DMA      [mem 0x0000000040000000-0x0000000041ff6fff]
  Normal   empty
  HighMem  empty
Movable zone start for each node
Early memory node ranges
  node   0: [mem 0x0000000040000000-0x0000000041ff6fff]
Initmem setup node 0 [mem 0x0000000040000000-0x0000000041ff6fff]
OF stdout device is: /a::a
PROM: Built device tree with 52609 bytes of memory.
Booting Linux...
percpu: Embedded 10 pages/cpu s17708 r0 d23252 u40960
Built 1 zonelists, mobility grouping off.  Total pages: 8119
Kernel command line: console=ttyS0,38400
Dentry cache hash table entries: 4096 (order: 2, 16384 bytes, linear)
Inode-cache hash table entries: 2048 (order: 1, 8192 bytes, linear)
Sorting __ex_table...
mem auto-init: stack:all(zero), heap alloc:off, heap free:off
Memory: 21228K/32732K available (4297K kernel code, 202K rwdata, 648K rodata, 3388K ini
t, 274K bss, 11504K reserved, 0K cma-reserved, 0K highmem)
rcu: Hierarchical RCU implementation.
rcu:    RCU event tracing is enabled.
rcu:    RCU restricting CPUs from NR_CPUS=32 to nr_cpu_ids=2.
rcu: RCU calculated value of scheduler-enlistment delay is 10 jiffies.
rcu: Adjusting geometry for rcu_fanout_leaf=16, nr_cpu_ids=2
NR_IRQS: 64
Console: colour dummy device 80x25
```

```
printk: console [ttyS0] enabled
printk: console [ttyS0] enabled
printk: bootconsole [earlyprom0] disabled
printk: bootconsole [earlyprom0] disabled
clocksource: timer_cs: mask: 0xffffffffffffffff max_cycles: 0x1d854df40, max_idle_ns: 3
526361616960 ns
Calibrating delay loop... 99.22 BogoMIPS (lpj=496128)
pid_max: default: 32768 minimum: 301
Mount-cache hash table entries: 1024 (order: 0, 4096 bytes, linear)
Mountpoint-cache hash table entries: 1024 (order: 0, 4096 bytes, linear)
Entering SMP Mode...
leon: SMP IPIs at IRQ 13
0:(2:32) cpus mpirq at 0x80000210
leon: power management initialized
rcu: Hierarchical SRCU implementation.
smp: Bringing up secondary CPUs ...
Starting CPU 1 : (irqmp: 0x80000210)
Started CPU 1
smp: Brought up 1 node, 2 CPUs
Total of 2 processors activated (198.96 BogoMIPS).
devtmpfs: initialized
clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns: 19112604462
750000 ns
futex hash table entries: 512 (order: 2, 16384 bytes, linear)
NET: Registered protocol family 16
clocksource: Switched to clocksource timer_cs
FS-Cache: Loaded
NET: Registered protocol family 2
IP idents hash table entries: 2048 (order: 2, 16384 bytes, linear)
tcp_listen_portaddr_hash hash table entries: 512 (order: 0, 6144 bytes, linear)
TCP established hash table entries: 1024 (order: 0, 4096 bytes, linear)
TCP bind hash table entries: 1024 (order: 1, 8192 bytes, linear)
TCP: Hash tables configured (established 1024 bind 1024)
UDP hash table entries: 256 (order: 1, 8192 bytes, linear)
UDP-Lite hash table entries: 256 (order: 1, 8192 bytes, linear)
NET: Registered protocol family 1
RPC: Registered named UNIX socket transport module.
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
RPC: Registered tcp NFSv4.1 backchannel transport module.
workingset: timestamp_bits=30 max_order=13 bucket_order=0
FS-Cache: Netfs 'nfs' registered for caching
NFS: Registering the id_resolver key type
Key type id_resolver registered
Key type id_legacy registered
Block layer SCSI generic (bsg) driver version 0.4 loaded (major 254)
io scheduler mq-deadline registered
io scheduler kyber registered
ioremap: done with statics, switching to malloc
Serial: GRLIB APBUART driver
ffd11f1c: ttyS0 at MMIO 0x80000100 (irq = 14, base_baud = 6250000) is a GRLIB/APBUART
grlib-apbuart at 0x80000100, irq 14
ffd112c8: ttyS1 at MMIO 0x80100100 (irq = 25, base_baud = 6250000) is a GRLIB/APBUART
grlib-apbuart at 0x80100100, irq 25
ffd11200: ttyS2 at MMIO 0x80100200 (irq = 26, base_baud = 6250000) is a GRLIB/APBUART
grlib-apbuart at 0x80100200, irq 26
ffd11138: ttyS3 at MMIO 0x80100300 (irq = 27, base_baud = 6250000) is a GRLIB/APBUART
grlib-apbuart at 0x80100300, irq 27
ffd11070: ttyS4 at MMIO 0x80100400 (irq = 28, base_baud = 6250000) is a GRLIB/APBUART
grlib-apbuart at 0x80100400, irq 28
ffd10fa8: ttyS5 at MMIO 0x80100500 (irq = 29, base_baud = 6250000) is a GRLIB/APBUART
grlib-apbuart at 0x80100500, irq 29
NET: Registered protocol family 10
Segment Routing with IPv6
sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
NET: Registered protocol family 17
Key type dns_resolver registered
Freeing unused kernel memory: 3388K
This architecture does not have kernel memory protection.
Run /init as init process
Saving 256 bits of non-creditable seed for next boot
Starting syslogd: OK
Starting klogd: OK
Running sysctl: OK
Starting network: OK
Starting crond: OK

Welcome to Buildroot
buildroot login: random: crng init done
```

```
printk: console [ttyS0] enabled
printk: console [ttyS0] enabled
printk: bootconsole [earlyprom0] disabled
printk: bootconsole [earlyprom0] disabled
clocksource: timer_cs: mask: 0xffffffffffffffff max_cycles: 0x1d854df40, max_idle_ns: 3
526361616960 ns
Calibrating delay loop... 99.22 BogoMIPS (lpj=496128)
pid_max: default: 32768 minimum: 301
Mount-cache hash table entries: 1024 (order: 0, 4096 bytes, linear)
Mountpoint-cache hash table entries: 1024 (order: 0, 4096 bytes, linear)
Entering SMP Mode...
leon: SMP IPIs at IRQ 13
0:(2:32) cpus mpirq at 0x80000210
leon: power management initialized
rcu: Hierarchical SRCU implementation.
smp: Bringing up secondary CPUs ...
Starting CPU 1 : (irqmp: 0x80000210)
Started CPU 1
smp: Brought up 1 node, 2 CPUs
Total of 2 processors activated (198.96 BogoMIPS).
devtmpfs: initialized
clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns: 19112604462
750000 ns
futex hash table entries: 512 (order: 2, 16384 bytes, linear)
NET: Registered protocol family 16
clocksource: Switched to clocksource timer_cs
FS-Cache: Loaded
NET: Registered protocol family 2
IP idents hash table entries: 2048 (order: 2, 16384 bytes, linear)
tcp_listen_portaddr_hash hash table entries: 512 (order: 0, 6144 bytes, linear)
TCP established hash table entries: 1024 (order: 0, 4096 bytes, linear)
TCP bind hash table entries: 1024 (order: 1, 8192 bytes, linear)
TCP: Hash tables configured (established 1024 bind 1024)
UDP hash table entries: 256 (order: 1, 8192 bytes, linear)
UDP-Lite hash table entries: 256 (order: 1, 8192 bytes, linear)
NET: Registered protocol family 1
RPC: Registered named UNIX socket transport module.
RPC: Registered udp transport module.
RPC: Registered tcp transport module.
RPC: Registered tcp NFSv4.1 backchannel transport module.
workingset: timestamp_bits=30 max_order=13 bucket_order=0
FS-Cache: Netfs 'nfs' registered for caching
NFS: Registering the id_resolver key type
Key type id_resolver registered
Key type id_legacy registered
Block layer SCSI generic (bsg) driver version 0.4 loaded (major 254)
io scheduler mq-deadline registered
io scheduler kyber registered
ioremap: done with statics, switching to malloc
Serial: GRLIB APBUART driver
ffd11f1c: ttyS0 at MMIO 0x80000100 (irq = 14, base_baud = 6250000) is a GRLIB/APBUART
grlib-apbuart at 0x80000100, irq 14
ffd112c8: ttyS1 at MMIO 0x80100100 (irq = 25, base_baud = 6250000) is a GRLIB/APBUART
grlib-apbuart at 0x80100100, irq 25
ffd11200: ttyS2 at MMIO 0x80100200 (irq = 26, base_baud = 6250000) is a GRLIB/APBUART
grlib-apbuart at 0x80100200, irq 26
ffd11138: ttyS3 at MMIO 0x80100300 (irq = 27, base_baud = 6250000) is a GRLIB/APBUART
grlib-apbuart at 0x80100300, irq 27
ffd11070: ttyS4 at MMIO 0x80100400 (irq = 28, base_baud = 6250000) is a GRLIB/APBUART
grlib-apbuart at 0x80100400, irq 28
ffd10fa8: ttyS5 at MMIO 0x80100500 (irq = 29, base_baud = 6250000) is a GRLIB/APBUART
grlib-apbuart at 0x80100500, irq 29
NET: Registered protocol family 10
Segment Routing with IPv6
sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
NET: Registered protocol family 17
Key type dns_resolver registered
Freeing unused kernel memory: 3388K
This architecture does not have kernel memory protection.
Run /init as init process
Saving 256 bits of non-creditable seed for next boot
Starting syslogd: OK
Starting klogd: OK
Running sysctl: OK
Starting network: OK
Starting crond: OK

Welcome to Buildroot
buildroot login: random: crng init done
```

■ `-romsz` : `laysim-GR712RC` provides 8MB of MRAM by default, and the MRAM size can be changed using the `-romsz` option. The updated ROM size is reflected in `FTMCTRL.MCFG1.PZ`.

• 16 : 16KB, 32 : 32KB, 64 : 64KB, 128 : 128KB, 256 : 256KB, 512 : 512KB, 1024 : 1MB, 2048 : 2MB, 4096 : 4MB,

• 8192 : 8MB (default), 16384 : 16MB, 32768 : 32MB, 65536 : 64MB, 131072 : 128MB, 262144 : 256MB

■ `-freq` : `laysim-GR712RC` operates at 100MHz by default, and the operating frequency (in MHz) can be modified using the `-freq` option. The frequency can be set from a minimum of 20MHz up to a maximum of 100MHz, and performance differences can be observed depending on the selected frequency. The configured frequency is also applied to communication interfaces such as SpW, 1553B, CAN, and UART.

Figure 3-14. Using '-freq' start-up option

laysim-gr712rc-cli in Linux	laysim-gr712rc-cli in Windows

```
[layright@CentOS BCC-2.2.1]$ laysim-gr712rc-cli -r -core0 dhry.elf -freq 50
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - dhry.elf
   - .text, addr: 0x40000000, size 60160 bytes
   - .rodata, addr: 0x4000EB00, size 2048 bytes
   - .data, addr: 0x4000F300, size 1568 bytes
   - .bss, addr: 0x4000F920, size 10688 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
Execution starts, 1000000 runs through Dhrystone
Total execution time:                      11.5 s
Microseconds for one run through Dhrystone:   11.5
Dhrystones per Second:                     86956.3

Dhrystones MIPS      :                      4
CORE#0 halts because of Error Mode [578635030cycles/5.58097sec]
9.5

 Performance statistics for CORE#0
   Cycles       : 578635030
   Instructions : 360279201
   Overall CPI  : 1.61
   CORE#0 performance (50MHz)  : 31.13 MOPS (31.13 MIPS, 0.00 MFLOPS)
   Simulated time              : 11.57 s
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 207.36 %
   laysim-gr712rc performance  : 64.55 MIPS
   Wall clock                  : 5.58 s

[layright@CentOS BCC-2.2.1]$ laysim-gr712rc-cli -r -core0 dhry.elf -freq 80
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - dhry.elf
   - .text, addr: 0x40000000, size 60160 bytes
   - .rodata, addr: 0x4000EB00, size 2048 bytes
   - .data, addr: 0x4000F300, size 1568 bytes
   - .bss, addr: 0x4000F920, size 10688 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
Execution starts, 1000000 runs through Dhrystone
Total execution time:                       7.2 s
Microseconds for one run through Dhrystone:    7.2
Dhrystones per Second:                     139130.1

Dhrystones MIPS      :                      7
CORE#0 halts because of Error Mode [580811770cycles/5.61311sec]
9.2

 Performance statistics for CORE#0
   Cycles       : 580811770
   Instructions : 361041052
   Overall CPI  : 1.61
   CORE#0 performance (80MHz)  : 49.73 MOPS (49.73 MIPS, 0.00 MFLOPS)
   Simulated time              : 7.26 s
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 129.34 %
   laysim-gr712rc performance  : 64.32 MIPS
   Wall clock                  : 5.61 s

[layright@CentOS BCC-2.2.1]$ laysim-gr712rc-cli -r -core0 dhry.elf -freq 100
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - dhry.elf
   - .text, addr: 0x40000000, size 60160 bytes
   - .rodata, addr: 0x4000EB00, size 2048 bytes
   - .data, addr: 0x4000F300, size 1568 bytes
   - .bss, addr: 0x4000F920, size 10688 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
Execution starts, 1000000 runs through Dhrystone
Total execution time:                       5.8 s
Microseconds for one run through Dhrystone:    5.8
Dhrystones per Second:                     173912.6

Dhrystones MIPS      :                      9
CORE#0 halts because of Error Mode [582253328cycles/5.49816sec]
9.0

 Performance statistics for CORE#0
   Cycles       : 582253328
   Instructions : 361545533
   Overall CPI  : 1.61
   CORE#0 performance (100MHz) : 62.09 MOPS (62.09 MIPS, 0.00 MFLOPS)
   Simulated time              : 5.82 s
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 105.90 %
   laysim-gr712rc performance  : 65.76 MIPS
   Wall clock                  : 5.50 s
[layright@CentOS BCC-2.2.1]$
```

```
PS C:\laysim-gr712rc\BCC-2.2.1> laysim-gr712rc-cli -r -core0 dhry.elf -freq 50
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - dhry.elf
   - .text, addr: 0x40000000, size 60160 bytes
   - .rodata, addr: 0x4000EB00, size 2048 bytes
   - .data, addr: 0x4000F300, size 1568 bytes
   - .bss, addr: 0x4000F920, size 10688 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
Execution starts, 1000000 runs through Dhrystone
Total execution time:                      11.5 s
Microseconds for one run through Dhrystone:   11.5
Dhrystones per Second:                     86956.3

Dhrystones MIPS      :                      4
CORE#0 halts because of Error Mode [578635030cycles/6.10459sec]
9.5

 Performance statistics for CORE#0
   Cycles       : 578635030
   Instructions : 360279201
   Overall CPI  : 1.61
   CORE#0 performance (50MHz)  : 31.13 MOPS (31.13 MIPS, 0.00 MFLOPS)
   Simulated time              : 11.57 s
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 189.57 %
   laysim-gr712rc performance  : 59.02 MIPS
   Wall clock                  : 6.10 s

PS C:\laysim-gr712rc\BCC-2.2.1> laysim-gr712rc-cli -r -core0 dhry.elf -freq 80
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - dhry.elf
   - .text, addr: 0x40000000, size 60160 bytes
   - .rodata, addr: 0x4000EB00, size 2048 bytes
   - .data, addr: 0x4000F300, size 1568 bytes
   - .bss, addr: 0x4000F920, size 10688 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
Execution starts, 1000000 runs through Dhrystone
Total execution time:                       7.2 s
Microseconds for one run through Dhrystone:    7.2
Dhrystones per Second:                     139130.1

Dhrystones MIPS      :                      7
CORE#0 halts because of Error Mode [580811770cycles/6.1423sec]
9.2

 Performance statistics for CORE#0
   Cycles       : 580811770
   Instructions : 361041052
   Overall CPI  : 1.61
   CORE#0 performance (80MHz)  : 49.73 MOPS (49.73 MIPS, 0.00 MFLOPS)
   Simulated time              : 7.26 s
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 118.20 %
   laysim-gr712rc performance  : 58.78 MIPS
   Wall clock                  : 6.14 s

PS C:\laysim-gr712rc\BCC-2.2.1> laysim-gr712rc-cli -r -core0 dhry.elf -freq 100
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - dhry.elf
   - .text, addr: 0x40000000, size 60160 bytes
   - .rodata, addr: 0x4000EB00, size 2048 bytes
   - .data, addr: 0x4000F300, size 1568 bytes
   - .bss, addr: 0x4000F920, size 10688 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
Execution starts, 1000000 runs through Dhrystone
Total execution time:                       5.8 s
Microseconds for one run through Dhrystone:    5.8
Dhrystones per Second:                     173912.6

Dhrystones MIPS      :                      9
CORE#0 halts because of Error Mode [582253328cycles/5.93763sec]
9.0

 Performance statistics for CORE#0
   Cycles       : 582253328
   Instructions : 361545533
   Overall CPI  : 1.61
   CORE#0 performance (100MHz) : 62.09 MOPS (62.09 MIPS, 0.00 MFLOPS)
   Simulated time              : 5.82 s
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 98.06 %
   laysim-gr712rc performance  : 60.89 MIPS
   Wall clock                  : 5.94 s
PS C:\laysim-gr712rc\BCC-2.2.1>
```

■ `-baud` : When RAM code is loaded, `laysim-GR712RC` sets the default baudrate of APBUART#0 to 38400bps. The baudrate of APBUART#0 can be set using the `-baud` option (the `fast_uart` option is used separately).

Figure 3-15. Using '-baud' start-up option

laysim-gr712rc-cli in Linux	laysim-gr712rc-cli in Windows

`[layright@CentOS BCC-2.2.1]$ laysim-gr712rc-cli -r -core0 dhry.elf -baud 9600`
`laysim for GR712RC Dual Core Processor CLI v3.1 by layright`
 	
`CORE#0 RAM file is loaded - dhry.elf`
`   - .text, addr: 0x40000000, size 60160 bytes`
	`PS C:\laysim-gr712rc\BCC-2.2.1> laysim-gr712rc-cli -r -core0 dhry.elf -baud 9600`
	`laysim for GR712RC Dual Core Processor CLI v3.1 by layright`
	
	`CORE#0 RAM file is loaded - dhry.elf`
	`   - .text, addr: 0x40000000, size 60160 bytes`

```
   - .rodata, addr: 0x4000EB00, size 2048 bytes
   - .data, addr: 0x4000F300, size 1568 bytes
   - .bss, addr: 0x4000F920, size 10688 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
Execution starts, 1000000 runs through Dhrystone
Total execution time:                    5.8 s
Microseconds for one run through Dhrystone:    5.8
Dhrystones per Second:                 173912.6

Dhrystones MIPS      :                   9
CORE#0 halts because of Error Mode [603823758cycles/5.58406sec]
9.0

 Performance statistics for CORE#0
   Cycles       : 603823758
   Instructions : 369095185
   Overall CPI  : 1.64
   CORE#0 performance (100MHz) : 61.13 MOPS (61.13 MIPS, 0.00 MFLOPS)
   Simulated time              : 6.04 s
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 108.13 %
   laysim-gr712rc performance  : 66.10 MIPS
   Wall clock                  : 5.58 s

[layright@CentOS BCC-2.2.1]$ laysim-gr712rc-cli -r -core0 dhry.elf -baud 115200
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - dhry.elf
   - .text, addr: 0x40000000, size 60160 bytes
   - .rodata, addr: 0x4000EB00, size 2048 bytes
   - .data, addr: 0x4000F300, size 1568 bytes
   - .bss, addr: 0x4000F920, size 10688 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
Execution starts, 1000000 runs through Dhrystone
Total execution time:                    5.8 s
Microseconds for one run through Dhrystone:    5.8
Dhrystones per Second:                 173912.6

Dhrystones MIPS      :                   9
CORE#0 halts because of Error Mode [577459978cycles/5.50584sec]
9.0

 Performance statistics for CORE#0
   Cycles       : 577459978
   Instructions : 359867893
   Overall CPI  : 1.60
   CORE#0 performance (100MHz) : 62.32 MOPS (62.32 MIPS, 0.00 MFLOPS)
   Simulated time              : 5.77 s
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 104.88 %
   laysim-gr712rc performance  : 65.36 MIPS
   Wall clock                  : 5.51 s
[layright@CentOS BCC-2.2.1]$
```

```
   - .rodata, addr: 0x4000EB00, size 2048 bytes
   - .data, addr: 0x4000F300, size 1568 bytes
   - .bss, addr: 0x4000F920, size 10688 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
Execution starts, 1000000 runs through Dhrystone
Total execution time:                    5.8 s
Microseconds for one run through Dhrystone:    5.8
Dhrystones per Second:                 173912.6

Dhrystones MIPS      :                   9
CORE#0 halts because of Error Mode [603823758cycles/6.04839sec]
9.0

 Performance statistics for CORE#0
   Cycles       : 603823758
   Instructions : 369095185
   Overall CPI  : 1.64
   CORE#0 performance (100MHz) : 61.13 MOPS (61.13 MIPS, 0.00 MFLOPS)
   Simulated time              : 6.04 s
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 99.83 %
   laysim-gr712rc performance  : 61.02 MIPS
   Wall clock                  : 6.05 s

PS C:\laysim-gr712rc\BCC-2.2.1> laysim-gr712rc-cli -r -core0 dhry.elf -baud 115200
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - dhry.elf
   - .text, addr: 0x40000000, size 60160 bytes
   - .rodata, addr: 0x4000EB00, size 2048 bytes
   - .data, addr: 0x4000F300, size 1568 bytes
   - .bss, addr: 0x4000F920, size 10688 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
Execution starts, 1000000 runs through Dhrystone
Total execution time:                    5.8 s
Microseconds for one run through Dhrystone:    5.8
Dhrystones per Second:                 173912.6

Dhrystones MIPS      :                   9
CORE#0 halts because of Error Mode [577459978cycles/6.01669sec]
9.0


 Performance statistics for CORE#0
   Cycles       : 577459978
   Instructions : 359867893
   Overall CPI  : 1.60
   CORE#0 performance (100MHz) : 62.32 MOPS (62.32 MIPS, 0.00 MFLOPS)
   Simulated time              : 5.77 s
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 95.98 %
   laysim-gr712rc performance  : 59.81 MIPS
   Wall clock                  : 6.02 s
PS C:\laysim-gr712rc\BCC-2.2.1>
```

- `-quantum` : `laysim-gr712rc-dbt` uses the time quantum (TQ) method for multicore scheduling and synchronization. The default TQ is 200 (200 x 10nsec = 2usec), and it can be changed by '`-quantum` VALUE' on start. Increasing TQ improves RTP but may violate core synchronization because the synchronization occurs at every quantum boundary. When the time quantum is set 0 in `laysim-gr712rc-dbt`, it changes the multicore scheduling to the basic block scheduling.

Figure 3-16. Using '-quantum' start-up option

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS RTEMS5.1-SMP]$ laysim-gr712rc-dbt-cli -r -core0 smpatomic01.elf -quant
um 0
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Mutlicore control : Basic block
CORE#0 RAM file is loaded - smpatomic01.elf
   - .text, addr: 0x40000000, size 116352 bytes
   - .rtemsroset, addr: 0x4001C680, size 144 bytes
   - .rtemsstack, addr: 0x4001C740/0x4001C710, size 262144 bytes (not loaded, just inf
o.)
   - .data, addr: 0x4005C740, size 1328 bytes
   - .bss, addr: 0x4005CC80, size 109088 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

*** BEGIN OF TEST SMPATOMIC 1 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
=== atomic add test case ===
worker 0 value: 3103420
worker 1 value: 3105531
atomic value: expected = 6208951, actual = 6208951
=== atomic flag test case ===
worker 0 value: 2475193
worker 1 value: 2476879
atomic value: expected = 4952072, actual = 4952072
=== atomic sub test case ===
worker 0 value: 4291874203
worker 1 value: 4291872094
atomic value: expected = 4288779001, actual = 4288779001
=== atomic compare exchange test case ===
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\RTEMS5.1-SMP> laysim-gr712rc-dbt-cli -r -core0 smpatomic01.elf -qu
antum 0
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Mutlicore control : Basic block
CORE#0 RAM file is loaded - smpatomic01.elf
   - .text, addr: 0x40000000, size 116352 bytes
   - .rtemsroset, addr: 0x4001C680, size 144 bytes
   - .rtemsstack, addr: 0x4001C740/0x4001C710, size 262144 bytes (not loaded, just inf
o.)
   - .data, addr: 0x4005C740, size 1328 bytes
   - .bss, addr: 0x4005CC80, size 109088 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

*** BEGIN OF TEST SMPATOMIC 1 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
=== atomic add test case ===
worker 0 value: 3103420
worker 1 value: 3105531
atomic value: expected = 6208951, actual = 6208951
=== atomic flag test case ===
worker 0 value: 2475193
worker 1 value: 2476879
atomic value: expected = 4952072, actual = 4952072
=== atomic sub test case ===
worker 0 value: 4291874203
worker 1 value: 4291872094
atomic value: expected = 4288779001, actual = 4288779001
=== atomic compare exchange test case ===
```

```
worker 0 value: 2430810
worker 1 value: 2432455
atomic value: expected = 4863265, actual = 4863265
=== atomic or/and test case ===
worker 0 value: 1
worker 1 value: 0
atomic value: expected = 1, actual = 1
=== atomic fence test case ===
normal value = 6640710, second value = 6640710
=== atomic store release and load acquire test case ===
processor 0 delta 1000ns, load count 0
processor 1 delta 2000ns, load count 6
=== atomic read-modify-write test case ===
processor 0 delta 1000ns, read-modify-write count 0
processor 1 delta 2000ns, read-modify-write count 0
=== single writer seqlock test case ===
processor 0 count 3413680
processor 1 count 2107749
=== multi writer seqlock test case ===
processor 0 count 2359491
processor 1 count 2109915

*** END OF TEST SMPATOMIC 1 ***

CORE#1 halts because of Error Mode [0.858096sec/841281043, 841281019cycles]

 Performance statistics for CORE#0
   Cycles       : 841281043
   Instructions : 362289012
   CORE#0 performance (100MHz) : 43.06 MIPS
   Simulated time              : 8.41 s

 Performance statistics for CORE#1
   Cycles       : 841281019
   Instructions : 383372630
   CORE#1 performance (100MHz) : 45.57 MIPS
   Simulated time              : 8.41 s

 Performance of laysim-gr712rc
   Real-time performance       : 980.39 %
   laysim-gr712rc performance  : 868.96 MIPS
   Wall clock                  : 0.86 s

[layright@CentOS RTEMS5.1-SMP]$ laysim-gr712rc-dbt-cli -r -core0 smpatomic01.elf -quant
um 1000
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (1000)
CORE#0 RAM file is loaded - smpatomic01.elf
   - .text, addr: 0x40000000, size 116352 bytes
   - .rtemsroset, addr: 0x4001C680, size 144 bytes
   - .rtemsstack, addr: 0x4001C740/0x4001C710, size 262144 bytes (not loaded, just inf
o.)
   - .data, addr: 0x4005C740, size 1328 bytes
   - .bss, addr: 0x4005CC80, size 109088 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

*** BEGIN OF TEST SMPATOMIC 1 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
=== atomic add test case ===
worker 0 value: 3103341
worker 1 value: 3105462
atomic value: expected = 6208803, actual = 6208803
=== atomic flag test case ===
worker 0 value: 2475154
worker 1 value: 2476848
atomic value: expected = 4952002, actual = 4952002
=== atomic sub test case ===
worker 0 value: 4291874262
worker 1 value: 4291872143
atomic value: expected = 4288779109, actual = 4288779109
=== atomic compare exchange test case ===
worker 0 value: 2430760
worker 1 value: 2432413
atomic value: expected = 4863173, actual = 4863173
=== atomic or/and test case ===
worker 0 value: 1
worker 1 value: 0
atomic value: expected = 1, actual = 1
=== atomic fence test case ===
normal value = 6640668, second value = 6640668
=== atomic store release and load acquire test case ===
processor 0 delta 0ns, load count 0
processor 1 delta 10000000ns, load count 142778
=== atomic read-modify-write test case ===
processor 0 delta 1ns, read-modify-write count 0
processor 1 delta 1ns, read-modify-write count 0
=== single writer seqlock test case ===
processor 0 count 3413657
processor 1 count 2107741
=== multi writer seqlock test case ===
processor 0 count 2359454
processor 1 count 2109890

*** END OF TEST SMPATOMIC 1 ***

CORE#1 halts because of Error Mode [0.529744sec/841457886, 841457186cycles]

 Performance statistics for CORE#0
   Cycles       : 841457886
   Instructions : 363869958
   CORE#0 performance (100MHz) : 43.24 MIPS
   Simulated time              : 8.41 s

 Performance statistics for CORE#1
   Cycles       : 841457186
   Instructions : 385019873
   CORE#1 performance (100MHz) : 45.76 MIPS
   Simulated time              : 8.41 s

 Performance of laysim-gr712rc
   Real-time performance       : 1588.38 %
   laysim-gr712rc performance  : 1413.65 MIPS
   Wall clock                  : 0.53 s
[layright@CentOS RTEMS5.1-SMP]$
```

```
worker 0 value: 2430810
worker 1 value: 2432455
atomic value: expected = 4863265, actual = 4863265
=== atomic or/and test case ===
worker 0 value: 1
worker 1 value: 0
atomic value: expected = 1, actual = 1
=== atomic fence test case ===
normal value = 6640710, second value = 6640710
=== atomic store release and load acquire test case ===
processor 0 delta 1000ns, load count 0
processor 1 delta 2000ns, load count 6
=== atomic read-modify-write test case ===
processor 0 delta 1000ns, read-modify-write count 0
processor 1 delta 2000ns, read-modify-write count 0
=== single writer seqlock test case ===
processor 0 count 3413680
processor 1 count 2107749
=== multi writer seqlock test case ===
processor 0 count 2359491
processor 1 count 2109915

*** END OF TEST SMPATOMIC 1 ***

CORE#1 halts because of Error Mode [0.906231sec/841281043, 841281019cycles]

 Performance statistics for CORE#0
   Cycles       : 841281043
   Instructions : 362289012
   CORE#0 performance (100MHz) : 43.06 MIPS
   Simulated time              : 8.41 s

 Performance statistics for CORE#1
   Cycles       : 841281019
   Instructions : 383372630
   CORE#1 performance (100MHz) : 45.57 MIPS
   Simulated time              : 8.41 s

 Performance of laysim-gr712rc
   Real-time performance       : 928.33 %
   laysim-gr712rc performance  : 822.82 MIPS
   Wall clock                  : 0.91 s

PS C:\laysim-gr712rc\RTEMS5.1-SMP> laysim-gr712rc-dbt-cli -r -core0 smpatomic01.elf -qu
antum 1000
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (1000)
CORE#0 RAM file is loaded - smpatomic01.elf
   - .text, addr: 0x40000000, size 116352 bytes
   - .rtemsroset, addr: 0x4001C680, size 144 bytes
   - .rtemsstack, addr: 0x4001C740/0x4001C710, size 262144 bytes (not loaded, just inf
o.)
   - .data, addr: 0x4005C740, size 1328 bytes
   - .bss, addr: 0x4005CC80, size 109088 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

*** BEGIN OF TEST SMPATOMIC 1 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
=== atomic add test case ===
worker 0 value: 3103341
worker 1 value: 3105462
atomic value: expected = 6208803, actual = 6208803
=== atomic flag test case ===
worker 0 value: 2475154
worker 1 value: 2476848
atomic value: expected = 4952002, actual = 4952002
=== atomic sub test case ===
worker 0 value: 4291874262
worker 1 value: 4291872143
atomic value: expected = 4288779109, actual = 4288779109
=== atomic compare exchange test case ===
worker 0 value: 2430760
worker 1 value: 2432413
atomic value: expected = 4863173, actual = 4863173
=== atomic or/and test case ===
worker 0 value: 1
worker 1 value: 0
atomic value: expected = 1, actual = 1
=== atomic fence test case ===
normal value = 6640668, second value = 6640668
=== atomic store release and load acquire test case ===
processor 0 delta 0ns, load count 0
processor 1 delta 10000000ns, load count 142778
=== atomic read-modify-write test case ===
processor 0 delta 1ns, read-modify-write count 0
processor 1 delta 1ns, read-modify-write count 0
=== single writer seqlock test case ===
processor 0 count 3413657
processor 1 count 2107741
=== multi writer seqlock test case ===
processor 0 count 2359454
processor 1 count 2109890

*** END OF TEST SMPATOMIC 1 ***

CORE#1 halts because of Error Mode [0.641239sec/841457886, 841457186cycles]

 Performance statistics for CORE#0
   Cycles       : 841457886
   Instructions : 363869958
   CORE#0 performance (100MHz) : 43.24 MIPS
   Simulated time              : 8.41 s

 Performance statistics for CORE#1
   Cycles       : 841457186
   Instructions : 385019873
   CORE#1 performance (100MHz) : 45.76 MIPS
   Simulated time              : 8.41 s

 Performance of laysim-gr712rc
   Real-time performance       : 1312.24 %
   laysim-gr712rc performance  : 1167.88 MIPS
   Wall clock                  : 0.64 s
PS C:\laysim-gr712rc\RTEMS5.1-SMP>
```

■ `-mod` : Load loadable module (SPI, I2C, UART, GPIO, I/O, 1553B RT, SpW Node and CAN Node modules). A loadable module should be dynamic shared objects (`.so` file on Linux and `.dll` file on Windows) and should follow the loadable interfaces of `laysim-GR712RC`. Where its interfaces and source codes are only available for the professional version.

Figure 3-17. Using '-mod' start-up option

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS RCC-1.3.1]$ laysim-gr712rc-dbt-cli -r -core0 rtems-rmap.elf -mod ../Loadable/loadable-SpW-Nodes/laysim_user_spw_node_module1.so
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ../Loadable/loadable-SpW-Nodes/laysim_user_spw_node_module1.so

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - rtems-rmap.elf
   - .text, addr: 0x40000000, size 237664 bytes
   - .rtemsroset, addr: 0x4003A060, size 176 bytes
   - .rtemsstack, addr: 0x4003A140, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x4003C140, size 8288 bytes
   - .bss, addr: 0x4003E1C0, size 106640 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

#### To change configuration of this example, you can set a breakpoint
#### on label <rmap_conf_label> and modify the following global variables:
## remote_base_address: 0x40000000 /* Base address on Remote. This is the address to wh
ich the data is copied. */
## remote_dst_address: 0xfe /* SpW Destination address. */
## remote_dst_key: 0x00/* SpW Destination Key */
## source_src_address: 0x20 /* SpW Source address */
## rmap_dev_idx: 0 /* This parameter chooses which initiator GRSPW2 core is used */
## route_entry { nr; /* Number of addresses in dstadr array */, dstadr[16];/* Path Addr
esses */}
## routetab[ROUTE_TO]={2,{1,5,0, ...}
## routetab[ROUTE_FROM]={2,{1,9,0, ...}
## path_addressing: 0 /* Use path addressing (!=0) or not (0) */
## rmap_pkt_size: 992
## rmap_data_bytes: 4096 /* Data bytes to send over RMAP */
#### RTEMS RMAP example

Setting up SpaceWire router
Configuring Router
Failed to open SpW Router
Router_configure: Error -1
Failed router initialization.
Limiting to 1 SpaceWire devices
Initializing SpaceWire device 0
 After Link Start: 5

Test built for at least 1 AMBA ports.
Note that the SpW-links will not be used.
There are 6 GRSPW cores in the system.
System clock: 10000 us / tick


Starting GRSPW0: DMA Started Successfully
RMAP connection is working
Initializing data array


Starting transmission type 1: one packet at a time

RMAP write cycle completed in 0.000588 seconds
RMAP read cycle completed in 0.000590 seconds
Data bytes sent: 4096, Packet size: 992, Packet count: 5
Wrote 4096 bytes to target in 0.000588 seconds. WRITE RATE: 6802.723163 KiB/s
Read 4096 bytes from target in 0.000590 seconds. READ RATE: 6779.700359 KiB/s
Wrote 5 packets to target in 0.000588 seconds. WRITE RATE: 8503.403953 pkt/s
Read 5 packets from target in 0.000590 seconds. READ RATE: 8474.625448 pkt/s


EXAMPLE 1 SUCCESSFULLY COMPLETED.

Starting transmission type 2: all packets at a time

RMAP write cycle completed in 0.000474 seconds
RMAP read cycle completed in 0.000421 seconds
Data bytes sent: 4096, Packet size: 992, Packet count: 5
Wrote 4096 bytes to target in 0.000474 seconds. WRITE RATE: 8439.243461 KiB/s
Read 4096 bytes from target in 0.000421 seconds. READ RATE: 9501.467365 KiB/s
Wrote 5 packets to target in 0.000474 seconds. WRITE RATE: 10549.054326 pkt/s
Read 5 packets from target in 0.000421 seconds. READ RATE: 11876.834206 pkt/s


EXAMPLE 2 SUCCESSFULLY COMPLETED.

CORE#0 halts because of Error Mode [0.057859sec/78601961, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 78601961
   Instructions : 45350689
   CORE#0 performance (100MHz) : 57.70 MIPS
   Simulated time              : 786.02 ms

 Performance of laysim-gr712rc
   Real-time performance       : 1358.51 %
   laysim-gr712rc performance  : 783.81 MIPS
   Wall clock                  : 0.06 s
[layright@CentOS RCC-1.3.1]$
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\RCC-1.3.1> laysim-gr712rc-dbt-cli -r -core0 rtems-rmap.elf -mod ..\Loadable\loadable-SpW-Nodes\laysim_user_spw_node_module1-dbt-cli.dll
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ..\Loadable\loadable-SpW-Nodes\laysim_user_spw_node_module1-dbt-cli.dll

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - rtems-rmap.elf
   - .text, addr: 0x40000000, size 237664 bytes
   - .rtemsroset, addr: 0x4003A060, size 176 bytes
   - .rtemsstack, addr: 0x4003A140, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x4003C140, size 8288 bytes
   - .bss, addr: 0x4003E1C0, size 106640 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

#### To change configuration of this example, you can set a breakpoint
#### on label <rmap_conf_label> and modify the following global variables:
## remote_base_address: 0x40000000 /* Base address on Remote. This is the address to wh
ich the data is copied. */
## remote_dst_address: 0xfe /* SpW Destination address. */
## remote_dst_key: 0x00/* SpW Destination Key */
## source_src_address: 0x20 /* SpW Source address */
## rmap_dev_idx: 0 /* This parameter chooses which initiator GRSPW2 core is used */
## route_entry { nr; /* Number of addresses in dstadr array */, dstadr[16];/* Path Addr
esses */}
## routetab[ROUTE_TO]={2,{1,5,0, ...}
## routetab[ROUTE_FROM]={2,{1,9,0, ...}
## path_addressing: 0 /* Use path addressing (!=0) or not (0) */
## rmap_pkt_size: 992
## rmap_data_bytes: 4096 /* Data bytes to send over RMAP */
#### RTEMS RMAP example

Setting up SpaceWire router
Configuring Router
Failed to open SpW Router
Router_configure: Error -1
Failed router initialization.
Limiting to 1 SpaceWire devices
Initializing SpaceWire device 0
 After Link Start: 5

Test built for at least 1 AMBA ports.
Note that the SpW-links will not be used.
There are 6 GRSPW cores in the system.
System clock: 10000 us / tick


Starting GRSPW0: DMA Started Successfully
RMAP connection is working
Initializing data array


Starting transmission type 1: one packet at a time

RMAP write cycle completed in 0.000588 seconds
RMAP read cycle completed in 0.000590 seconds
Data bytes sent: 4096, Packet size: 992, Packet count: 5
Wrote 4096 bytes to target in 0.000588 seconds. WRITE RATE: 6802.723163 KiB/s
Read 4096 bytes from target in 0.000590 seconds. READ RATE: 6779.700359 KiB/s
Wrote 5 packets to target in 0.000588 seconds. WRITE RATE: 8503.403953 pkt/s
Read 5 packets from target in 0.000590 seconds. READ RATE: 8474.625448 pkt/s


EXAMPLE 1 SUCCESSFULLY COMPLETED.

Starting transmission type 2: all packets at a time

RMAP write cycle completed in 0.000474 seconds
RMAP read cycle completed in 0.000421 seconds
Data bytes sent: 4096, Packet size: 992, Packet count: 5
Wrote 4096 bytes to target in 0.000474 seconds. WRITE RATE: 8439.243461 KiB/s
Read 4096 bytes from target in 0.000421 seconds. READ RATE: 9501.467365 KiB/s
Wrote 5 packets to target in 0.000474 seconds. WRITE RATE: 10549.054326 pkt/s
Read 5 packets from target in 0.000421 seconds. READ RATE: 11876.834206 pkt/s


EXAMPLE 2 SUCCESSFULLY COMPLETED.

CORE#0 halts because of Error Mode [0.250031sec/78601961, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 78601961
   Instructions : 45350689
   CORE#0 performance (100MHz) : 57.70 MIPS
   Simulated time              : 786.02 ms

 Performance of laysim-gr712rc
   Real-time performance       : 314.37 %
   laysim-gr712rc performance  : 181.38 MIPS
   Wall clock                  : 0.25 s
PS C:\laysim-gr712rc\RCC-1.3.1>
```

■ `-ramws` : Set SRAM read/write waitstates (default 2 w/s). Also, it is possible to write `MCFG2` register in the control console using `wmem` command.

Figure 3-18. Using '-ramws' start-up option

laysim-gr712rc-cli in Linux

```
[layright@CentOS BCC-2.2.1]$ laysim-gr712rc-cli -r -core0 dhry.elf -ramws 0
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - dhry.elf
   - .text, addr: 0x40000000, size 60160 bytes
   - .rodata, addr: 0x4000EB00, size 2048 bytes
   - .data, addr: 0x4000F300, size 1568 bytes
   - .bss, addr: 0x4000F920, size 10688 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
Execution starts, 1000000 runs through Dhrystone
Total execution time:                          5.2 s
Microseconds for one run through Dhrystone:    5.2
Dhrystones per Second:                    191938.3

Dhrystones MIPS      :                      10
CORE#0 halts because of Error Mode [528243003cycles/5.47877sec]
9.2

 Performance statistics for CORE#0
   Cycles       : 528243003
   Instructions : 361546048
   Overall CPI  : 1.46
   CORE#0 performance (100MHz) : 68.44 MOPS (68.44 MIPS, 0.00 MFLOPS)
   Simulated time              : 5.28 s
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 96.42 %
   laysim-gr712rc performance  : 65.99 MIPS
   Wall clock                  : 5.48 s

[layright@CentOS BCC-2.2.1]$ laysim-gr712rc-cli -r -core0 dhry.elf -ramws 3
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - dhry.elf
   - .text, addr: 0x40000000, size 60160 bytes
   - .rodata, addr: 0x4000EB00, size 2048 bytes
   - .data, addr: 0x4000F300, size 1568 bytes
   - .bss, addr: 0x4000F920, size 10688 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
Execution starts, 1000000 runs through Dhrystone
Total execution time:                          6.0 s
Microseconds for one run through Dhrystone:    6.0
Dhrystones per Second:                    165837.0

Dhrystones MIPS      :                       9
CORE#0 halts because of Error Mode [610259458cycles/5.47344sec]
4.4

 Performance statistics for CORE#0
   Cycles       : 610259458
   Instructions : 361545132
   Overall CPI  : 1.69
   CORE#0 performance (100MHz) : 59.24 MOPS (59.24 MIPS, 0.00 MFLOPS)
   Simulated time              : 6.10 s
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 111.49 %
   laysim-gr712rc performance  : 66.05 MIPS
   Wall clock                  : 5.47 s
[layright@CentOS BCC-2.2.1]$
```

laysim-gr712rc-cli in Windows

```
PS C:\laysim-gr712rc\BCC-2.2.1> laysim-gr712rc-cli -r -core0 dhry.elf -ramws 0
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - dhry.elf
   - .text, addr: 0x40000000, size 60160 bytes
   - .rodata, addr: 0x4000EB00, size 2048 bytes
   - .data, addr: 0x4000F300, size 1568 bytes
   - .bss, addr: 0x4000F920, size 10688 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
Execution starts, 1000000 runs through Dhrystone
Total execution time:                          5.2 s
Microseconds for one run through Dhrystone:    5.2
Dhrystones per Second:                    191938.3

Dhrystones MIPS      :                      10
CORE#0 halts because of Error Mode [528243003cycles/6.09352sec]
9.2

 Performance statistics for CORE#0
   Cycles       : 528243003
   Instructions : 361546048
   Overall CPI  : 1.46
   CORE#0 performance (100MHz) : 68.44 MOPS (68.44 MIPS, 0.00 MFLOPS)
   Simulated time              : 5.28 s
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 86.69 %
   laysim-gr712rc performance  : 59.33 MIPS
   Wall clock                  : 6.09 s

PS C:\laysim-gr712rc\BCC-2.2.1> laysim-gr712rc-cli -r -core0 dhry.elf -ramws 3
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - dhry.elf
   - .text, addr: 0x40000000, size 60160 bytes
   - .rodata, addr: 0x4000EB00, size 2048 bytes
   - .data, addr: 0x4000F300, size 1568 bytes
   - .bss, addr: 0x4000F920, size 10688 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
Execution starts, 1000000 runs through Dhrystone
Total execution time:                          6.0 s
Microseconds for one run through Dhrystone:    6.0
Dhrystones per Second:                    165837.0

Dhrystones MIPS      :                       9
CORE#0 halts because of Error Mode [610259458cycles/6.06147sec]
4.4

 Performance statistics for CORE#0
   Cycles       : 610259458
   Instructions : 361545132
   Overall CPI  : 1.69
   CORE#0 performance (100MHz) : 59.24 MOPS (59.24 MIPS, 0.00 MFLOPS)
   Simulated time              : 6.10 s
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 100.68 %
   laysim-gr712rc performance  : 59.65 MIPS
   Wall clock                  : 6.06 s
PS C:\laysim-gr712rc\BCC-2.2.1>
```

■ `-air` : Configuration of GRETH and GRCAN for AIR hypervisor testsuites.

■ `-cfs` : Configuration of GRETH for NASA cFS with IP 10.10.10.22 and UDP port 1234/1235 for telecommand/telemetry.

■ `-gdb` : Listen for GDB connection directly at start-up. This option is only available in the CLI version of `laysim-GR712RC`.

■ `-port` : Set the port number to be used fog GDB communication. The default port number is 1234. This option is only available in the CLI version of `laysim-GR712RC`.

3.3 Control Console of laysim-GR712RC

The control console of `laysim-GR712RC` provides a variety of command line interfaces for controlling `laysim-GR712RC` described below.

■ `help` : The help command shows available commands in the control console as shown in Figure 3-19.

Figure 3-19. help command

help command in laysim-gr712rc

```
laysim> help

Execute help
 dis/display [addr]/[sym] [count]   : Disassemble [count] instructions at [addr]/[sym]
 mem/memory [addr]/[sym] [count]    : Show memory at [addr]/[sym] with [count]
 wmem [addr]/[sym] [value]          : Write a word to memory at [addr]/[sym] with [value]
 memdump [addr]/[sym] [wcnt] [file] : Save memory at [addr]/[sym] with [wcnt] to [file]
 aload [file]                       : Load additional ELF file without symbol/debugging
                                      - Only a ELF file can be loaded
 save [file]                        : Save laysim-gr712rc state to [file]
                                      - It also saves the state of loadable modules
 restore [file]                     : Restore laysim-gr712rc state from [file]
                                      - It also restores the state of loadable modules
 reg/register                       : Show GR712RC SPARC IU registers
 float                              : Show GR712RC SPARC FPU registers
 s/step                             : Single step
 n/next                             : Single step with showing IU registers
 t/trace [count]                    : Trace with [count]
 r/run                              : Run or continue
 stop                               : Stop execution
 reset                              : Reset laysim-GR712RC state on stop mode
 b/break [addr]/[sym]               : Add a breakpoint at [addr]/[sym]
 d/del [addr]/[sym]                 : Delete a breakpoint at [addr]/[sym]
 watch [addr]/[sym]                 : Add a watchpoint at [addr]/[sym]
 dw/dwatch [addr]/[sym]             : Delete a watchpoint at [addr]/[sym]
 icache [cmd]                       : GR712RC instruction cache operation
  : icache [core] disable             - Disable CORE# icache
  : icache [core] enable              - Enable  CORE# icache
  : icache [core] frozen              - Freeze  CORE# icache
  : icache [core] flush               - Flush   CORE# icache
  : icache [core] ite                 - Set CORE# ite in icache
  : icache [core] ide                 - Set CORE# ide in icache
 dcache [cmd]                       : GR712RC data cache operation
  : dcache [core] disable             - Disable CORE# dcache
  : dcache [core] enable              - Enable  CORE# dcache
  : dcache [core] frozen              - Freeze  CORE# dcache
  : dcache [core] flush               - Flush   CORE# dcache
  : dcache [core] dte                 - Set CORE# dte in dcache
  : dcache [core] dde                 - Set CORE# dde in dcache
 core [cmd]                         : CORE status and debug selection
  : core                              - Show status of cores
  : core 0                            - CORE#0 is selected for debugging
  : core 1                            - CORE#1 is selected for debugging
  : core 2                            - All COREs are selected for debugging
 history [core]                     : Show executed instructions (Max 1000)
 perf                               : Show performance statistics
  : perf reset                        - Reset performance statistics
 fastuart                           : Set APBUART#0 baudrate
  : fastuart on                       - Set APBUART#0 baudrate as infinite
  : fastuart off                      - Set APBUART#0 baudrate as correct
 regupdate                          : Set GR712RC APB register update in GUI in real-time
  : regupdate on                      - Set APB register update in GUI
  : regupdate off                     - Set no APB register update in GUI
 rtems                              : Show RTEMS v4.8/4.10/5.1 information
  : rtems task                        - Show RTEMS task information
  : rtems task [id]                   - Show RTEMS selected task information
  : rtems trap                        - Show RTEMS trap information
  : rtems interrupt                   - Show RTEMS BSP interrupt information (v4.10/5.1)
  : rtems core [num]                  - Show RTEMS5-SMP core [num] information
 mmu [core]                         : Show MMU registers for selected core
  : mmu 0                             - Show MMU registers for CORE#0
  : mmu 1                             - Show MMU registers for CORE#1
 walk [core] vaddr                  : Printout a MMU table walk for the given vaddr
  : walk 0 0xFC000000                 - Show vaddr to paddr of CORE#0
  : walk 1 0xFC000100                 - Show vaddr to paddr of CORE#1
 vdis [core] [vaddr] [count]        : Disassemble [count] instructions at [vaddr]
  : vdis 0 0xFC009350 0x10            - Disassmble instructions of CORE#0 at virtual address
 air                                : Show AIR/GMV hypervisor information
  : air                               - Show AIR/GMV information
  : air break                         - Show AIR partition breakpoints
  : air break [part] [addr]/[sym]     - Add a breakpoint at AIR partition
  : air del [part] [addr]/[sym]       - Delete a breakpoint at AIR partition
  : air pload [part] file             - Load the debugging information from AIR partition file
 xng-setup [addr]                   : Set up XNG RAM execution with XCF address
 event                              : Show events in the event queue
 info sys                           : Show GR712RC system information
 info reg [core_name]               : Show all registers of [core_name]
```

help command in laysim-gr712rc-dbt

```
laysim> help

Execute help
 dis/display [addr]/[sym] [count]   : Disassemble [count] instructions at [addr]/[sym]
 mem/memory [addr]/[sym] [count]    : Show memory at [addr]/[sym] with [count]
 wmem [addr]/[sym] [value]          : Write a word to memory at [addr]/[sym] with [value]
 aload [file]                       : Load additional ELF file without symbol/debugging
                                      - Only a ELF file can be loaded
 save [file]                        : Save laysim-gr712rc state to [file]
                                      - It also saves the state of loadable modules
 restore [file]                     : Restore laysim-gr712rc state from [file]
                                      - It also restores the state of loadable modules
 reg/register                       : Show GR712RC SPARC IU registers
 float                              : Show GR712RC SPARC FPU registers
 s/step                             : Single step
 n/next                             : Single step with showing IU registers
 t/trace [count]                    : Trace with [count]
 r/run                              : Run or continue
 stop                               : Stop execution
 reset                              : Reset laysim-GR712RC state on stop mode
 b/break [addr]/[sym]               : Add a breakpoint at [addr]/[sym]
  : break                             - Show all breakpoints
  : break Init                        - Set a breakpoint at Init
 d/del [addr]/[sym]                 : Delete a breakpoint at [addr]/[sym]
 watch [addr]/[sym]                 : Add a watchpoint at [addr]/[sym]
  : watch                             - Show all watchpoints
  : watch 0xFF900000                  - Set a write-watchpoint at 0xFF900000
 dw/dwatch [addr]/[sym]             : Delete a watchpoint at [addr]/[sym]
 core [cmd]                         : CORE status and debug selection
  : core                              - Show status of cores
  : core 0                            - CORE#0 is selected for debugging
  : core 1                            - CORE#1 is selected for debugging
  : core 2                            - All COREs are selected for debugging
 wakeup [core]                      : Wake-up CORE#n with [core]
 perf                               : Show performance statistics
 fastuart                           : Set UART baudrate
  : fastuart on                       - Set UART baudrate as infinite
  : fastuart off                      - Set UART baudrate as correct
 regupdate                          : Set GR712RC register update in GUI in real-time
  : regupdate on                      - Set register update in GUI
  : regupdate off                     - Set no register update in GUI
 rtems                              : Show RTEMS v4.8/4.10/5.1(SMP only) information
  : rtems task                        - Show RTEMS task information
  : rtems task [id]                   - Show RTEMS selected task information
  : rtems trap                        - Show RTEMS trap information
  : rtems interrupt                   - Show RTEMS BSP interrupt information (v4.10/5.1)
  : rtems core [num]                  - Show RTEMS5-SMP core [num] information
 mmu [core]                         : Show MMU registers for selected core
  : mmu 0                             - Show MMU registers for CORE#0
  : mmu 1                             - Show MMU registers for CORE#1
 walk [core] vaddr                  : Printout a MMU table walk for the given vaddr
  : walk 0 0xFC000000                 - Show vaddr to paddr of CORE#0
  : walk 1 0xFC000100                 - Show vaddr to paddr of CORE#1
 vdis [core] [vaddr] [count]        : Disassemble [count] instructions at [vaddr]
  : vdis 0 0xFC009350 0x10            - Disassmble instructions of CORE#0 at virtual address
 air                                : Show AIR/GMV hypervisor information
 xng-setup [addr]                   : Set up XNG RAM execution with XCF address
 event                              : Show events in the event queue
 info sys                           : Show GR712RC system information
 info reg [core_name]               : Show all registers of [core_name]
```

help command in laysim-gr712rc-cli

```
laysim> help

Execute help
 dis/display [addr]/[sym] [count]   : Disassemble [count] instructions at [addr]/[sym]
 mem/memory [addr]/[sym] [count]    : Show memory at [addr]/[sym] with [count]
 wmem [addr]/[sym] [value]          : Write a word to memory at [addr]/[sym] with [value]
 memdump [addr]/[sym] [wcnt] [file] : Save memory at [addr]/[sym] with [wcnt] to [file]
 aload [file]                       : Load additional ELF file without symbol/debugging
                                      - Only a ELF file can be loaded
 load [file]                        : Load GR712RC RAM [file] to CORE#0
                                      - Only a file can be loaded once
 batch [file]                       : Execute a batch [file] of laysim commands
 save [file]                        : Save laysim-gr712rc state to [file]
                                      - It also saves the state of loadable modules
 restore [file]                     : Restore laysim-gr712rc state from [file]
                                      - It also restores the state of loadable modules
 reg/register                       : Show GR712RC SPARC IU registers
 float                              : Show GR712RC SPARC FPU registers
 s/step                             : Single step
 n/next                             : Single step with showing IU registers
 t/trace [count]                    : Trace with [count]
 r/run                              : Run or continue
 rundelta [dcycles]                 : Run till current cycles + [dcycles]
 reset                              : Reset laysim-GR712RC state on stop mode
 b/break [addr]/[sym]               : Add a breakpoint at [addr]/[sym]
 d/del [addr]/[sym]                 : Delete a breakpoint at [addr]/[sym]
```

help command in laysim-gr712rc-dbt-cli

```
laysim> help

Execute help
 dis/display [addr]/[sym] [count]   : Disassemble [count] instructions at [addr]/[sym]
 mem/memory [addr]/[sym] [count]    : Show memory at [addr]/[sym] with [count]
 wmem [addr]/[sym] [value]          : Write a word to memory at [addr]/[sym] with [value]
 aload [file]                       : Load additional ELF file without symbol/debugging
                                      - Only a ELF file can be loaded
 memdump [addr]/[sym] [wcnt] [file] : Save memory at [addr]/[sym] with [wcnt] to [file]
 load [file]                        : Load GR712RC RAM [file] to CORE#0 on stop/reset mode
                                      - Only a file can be loaded once
 batch [file]                       : Execute a batch [file] of laysim commands
 save [file]                        : Save laysim-gr712rc state to [file]
                                      - It also saves the state of loadable modules
 restore [file]                     : Restore laysim-gr712rc state from [file]
                                      - It also restores the state of loadable modules
 reg/register                       : Show GR712RC SPARC IU registers
 float                              : Show GR712RC SPARC FPU registers
 s/step                             : Single step
 n/next                             : Single step with showing IU registers
 t/trace [count]                    : Trace with [count]
 r/run                              : Run or continue
 reset                              : Reset laysim-GR712RC state on stop mode
 b/break [addr]/[sym]               : Add a breakpoint at [addr]/[sym]
  : break                             - Show all breakpoints
  : break Init                        - Set a breakpoint at Init
```

```
watch [addr]/[sym]              : Add a watchpoint at [addr]/[sym]
dw/dwatch [addr]/[sym]          : Delete a watchpoint at [addr]/[sym]
icache [cmd]                    : GR712RC instruction cache operation
 : icache [core] disable          - Disable CORE# icache
 : icache [core] enable           - Enable  CORE# icache
 : icache [core] frozen           - Freeze  CORE# icache
 : icache [core] flush            - Flush   CORE# icache
 : icache [core] ite              - Set CORE# ite in icache
 : icache [core] ide              - Set CORE# ide in icache
 : icache [core] dump             - Dump CORE#0/#1 icache
dcache [cmd]                    : GR712RC data cache operation
 : dcache [core] disable          - Disable CORE# dcache
 : dcache [core] enable           - Enable  CORE# dcache
 : dcache [core] frozen           - Freeze  CORE# dcache
 : dcache [core] flush            - Flush   CORE# dcache
 : dcache [core] dte              - Set CORE# dte in dcache
 : dcache [core] dde              - Set CORE# dde in dcache
 : dcache [core] dump             - Dump CORE# dcache
core [cmd]                      : CORE status and debug selection
 : core                           - Show status of cores
 : core 0                         - CORE#0 is selected for debugging
 : core 1                         - CORE#1 is selected for debugging
 : core 2                         - All COREs are selected for debugging
history [core]                  : Show executed instructions (Max 1000)
perf                            : Show performance statistics
 : perf reset                     - Reset performance statistics
fastuart                        : Set APBUART#0 baudrate
 : fastuart on                    - Set APBUART#0 baudrate as infinite
 : fastuart off                   - Set APBUART#0 baudrate as correct
rtems                           : Show RTEMS v4.8/4.10/5.1 information
 : rtems task                     - Show RTEMS task information
 : rtems task [id]                - Show RTEMS selected task information
 : rtems trap                     - Show RTEMS trap information
 : rtems interrupt                - Show RTEMS BSP interrupt information (v4.10/5.1)
 : rtems core [num]               - Show RTEMS5-SMP core [num] information
mmu [core]                      : Show MMU registers for selected core
 : mmu 0                          - Show MMU registers for CORE#0
 : mmu 1                          - Show MMU registers for CORE#1
walk [core] vaddr               : Printout a MMU table walk for the given vaddr
 : walk 0 0xFC000000              - Show vaddr to paddr of CORE#0
 : walk 1 0xFC000100              - Show vaddr to paddr of CORE#1
vdis [core] [vaddr] [count]     : Disassemble [count] instructions at [vaddr]
 : vdis 0 0xFC009350 0x10         - Disassmble instructions of CORE#0 at virtual address
air                             : Show AIR/GMV hypervisor information
 : air                            - Show AIR/GMV information
 : air break                      - Show AIR partition breakpoints
 : air break [part] [addr]/[sym]  - Add a breakpoint at AIR partition
 : air del [part] [addr]/[sym]    - Delete a breakpoint at AIR partition
 : air pload [part] file          - Load the debugging information from AIR partition file
xng-setup [addr]                : Set up XNG RAM execution with XCF address
event                           : Show events in the event queue
info sys                        : Show GR712RC system information
info reg [core_name]            : Show all registers of [core_name]
gdb                             : Start GDB server
quit                            : Quit laysim CLI
```

```
d/del [addr]/[sym]              : Delete a breakpoint at [addr]/[sym]
watch [addr]/[sym]              : Add a watchpoint at [addr]/[sym]
 : watch                          - Show all watchpoints
 : watch 0xFF900000               - Set a write-watchpoint at 0xFF900000
dw/dwatch [addr]/[sym]          : Delete a watchpoint at [addr]/[sym]
core [cmd]                      : CORE status and debug selection
 : core                           - Show status of cores
 : core 0                         - CORE#0 is selected for debugging
 : core 1                         - CORE#1 is selected for debugging
 : core 2                         - All COREs are selected for debugging
wakeup [core]                   : Wake-up CORE#n with [core]
perf                            : Show performance statistics
fastuart                        : Set UART baudrate
 : fastuart on                    - Set UART baudrate as infinite
 : fastuart off                   - Set UART baudrate as correct
rtems                           : Show RTEMS v4.8/4.10/5.1(SMP only) information
 : rtems task                     - Show RTEMS task information
 : rtems task [id]                - Show RTEMS selected task information
 : rtems trap                     - Show RTEMS trap information
 : rtems interrupt                - Show RTEMS BSP interrupt information (v4.10/5.1)
 : rtems core [num]               - Show RTEMS5-SMP core [num] information
mmu [core]                      : Show MMU registers for selected core
 : mmu 0                          - Show MMU registers for CORE#0
 : mmu 1                          - Show MMU registers for CORE#1
walk [core] vaddr               : Printout a MMU table walk for the given vaddr
 : walk 0 0xFC000000              - Show vaddr to paddr of CORE#0
 : walk 1 0xFC000100              - Show vaddr to paddr of CORE#1
vdis [core] [vaddr] [count]     : Disassemble [count] instructions at [vaddr]
 : vdis 0 0xFC009350 0x10         - Disassmble instructions of CORE#0 at virtual address
air                             : Show AIR/GMV hypervisor information
xng-setup [addr]                : Set up XNG RAM execution with XCF address
event                           : Show events in the event queue
info sys                        : Show GR712RC system information
info reg [core_name]            : Show all registers of [core_name]
gdb                             : Start GDB server
quit                            : Quit laysim CLI
```

■ `display/dis` : The display command disassembles [count] instructions at [addr]/[sym] with [count] based on selected core as shown in Figure 3-20. Only SRAM/MRAM memory can be displayed . Default value for count is 10.

Figure 3-20. display command

```
laysim> dis main

[CORE#0] Address of main is 0x400013C8
 400013C8 : 9DE3BF48   save    %sp, -184, %sp
 400013CC : 40000621   call    0x40002c50
 400013D0 : 90102030   or      %g0, 0x30, %o0
 400013D4 : 03100048   sethi   %hi(0x40012000), %g1
 400013D8 : BA100008   or      %g0, %o0, %i5
 400013DC : FA206150   st      %i5, [%g1 + 0x150]
 400013E0 : 4000061C   call    0x40002c50
 400013E4 : 90102030   or      %g0, 0x30, %o0
 400013E8 : 82102002   or      %g0, 0x2, %g1
 400013EC : FA220000   st      %i5, [%o0]
```

■ `memory/mem` : The memory command shows the contents of AMBA memory at [addr]/[sym] with [count] based on selected core as shown in Figure 3-21. Default value for count is 10.

Figure 3-21. memory command

```
laysim> mem printf 40

[CORE#0] Address of printf is 0x40003828
 40003820 : 81C7E008 91E80008 9DE3BF90 F227A048     ............'.H
 40003830 : 01000000 0310003C F427A04C F627A050     .......<.'.L.'.P
 40003840 : F827A054 FA27A058 D0006338 80A22000     .'.T.'.X..c8.. .
 40003850 : 22800007 D2022008 C2022018 80A06000     "..... ... ...`.
```

```
  40003860 : 02800009 01000000 D2022008 9607A048     ......... ....H
  40003870 : D627BFFC 400001DA 94100018 81C7E008     .'..@...........
  40003880 : 91E80008 D027BFF4 7FFFFBA8 01000000     .....'..........
  40003890 : D007BFF4 D2022008 9607A048 D627BFFC     ...... ....H.'..
  400038A0 : 400001CF 94100018 81C7E008 91E80008     @...............
  400038B0 : 9DE3BFA0 80A62000 22800007 F4062008     ...... ."..... .

laysim> mem 0x80000000
  80000000 : 000280FF 7C40146A 08000000 00000000     ....|@.j........
  80000010 : 00000000 00000000 00000000 00000000     ................
  80000020 : 00000000 00000000 00000000 00000000     ................
```

■ `register/reg` : The register command shows the current status of selected core GR712RC IU registers as shown in Figure 3-22. There is no modification function of IU registers.

Figure 3-22. register command

```
laysim> core 2
GR712RC all COREs are selected for debugging

laysim> reg

 ---------------------- CORE#0 ----------------------          ---------------------- CORE#1 ----------------------
         INS       LOCALS       OUTS       GLOBALS                  INS       LOCALS       OUTS       GLOBALS
  0:  400D3CC0     4006C5D0     400D3DCC     00000000           0:  4006F348     4006F370     400D26D0     00000000
  1:  F34000E2     400D3DD8     4006DB28     0000CDE9           1:  00000007     400DC334     400DC334     0000CABB
  2:  400D3DCC     400D3DC8     00000000     0000CDE8           2:  400D3EC0     00000000     4006F2C0     0000CABB
  3:  400D3CC0     400D3DD4     00000005     400A8D78           3:  4006F0E8     4006F124     4006F104     00000000
  4:  00000000     400583A0     400D26E0     00000000           4:  400168D8     4006F090     4006F208     33333333
  5:  400D3DD8     4005839C     4006F348     00000000           5:  4006F080     00000000     00000000     00000000
  6:  400DA2B0     00000007     400DA230     400D3CC0           6:  400DC3E8     4006F0A8     400DC2C0     400D3EC0
  7:  40006F74     00000000     40009138     00000000           7:  40001330     4006F2C0     40007398     00000000

 psr:F3000FE2 wim:00000020 tbr:40000890 y:00001674          psr:F3400FE7 wim:00000002 tbr:40000890 y:00000000
 [n:0 z:0 v:0 c:0 ef:0 pil:15 s:1 ps:1 et:1 cwp:2]          [n:0 z:1 v:0 c:0 ef:0 pil:15 s:1 ps:1 et:1 cwp:7]

 pc:   40009248 : 02800035   be      0x4000931c            pc:   40011958 : 82006001   add     %g1, 0x1, %g1
 npc:  4000924C : 01000000   nop                          npc:  4001195C : 01000000   nop
```

■ `wmem` : The wmem command writes a 32bit word to AMBA memory at [addr]/[sym] with [value] as shown in Figure 3-23.

Figure 3-23. wmem command

```
laysim> wmem 0x8000000C 0x123

laysim> mem 0x80000000
  80000000 : 000280FF 7C40146A 08000000 00000123     ....|@.j.......#
  80000010 : 00000000 00000000 00000000 00000000     ................
  80000020 : 00000000 00000000 00000000 00000000     ................
```

■ `float` : The float command shows the current status of selected core of GR712RC FPU registers as shown in Figure 3-24. There is no modification function of FPU registers.

Figure 3-24. float command

```
laysim> core 2
GR712RC all COREs are selected for debugging

laysim> float

 ---------------------- CORE#0 ------------------------------      ---------------------- CORE#1 ------------------------------
 fsr: 00040820                                                    fsr: 00040000
 [nvm:0 ofm:0 ufm:0 dzm:0 nxm:0 ftt:0 fcc:2]                      [nvm:0 ofm:0 ufm:0 dzm:0 nxm:0 ftt:0 fcc:0]
 [nva:0 ofa:0 ufa:0 dza:0 nxa:1 nvc:0 ofc:0 ufc:0 dzc:0 nxc:0]    [nva:0 ofa:0 ufa:0 dza:0 nxa:0 nvc:0 ofc:0 ufc:0 dzc:0 nxc:0]
 f00   3F471868      7.777162e-01      7.048142e-04              f00   00000000      0.000000e+00      0.000000e+00
 f01   F20D2269     -2.795455e+30                                f01   00000000      0.000000e+00
 f02   00000000      0.000000e+00      0.000000e+00              f02   00000000      0.000000e+00      0.000000e+00
 f03   00000000      0.000000e+00                                f03   00000000      0.000000e+00
 f04   00000000      0.000000e+00      0.000000e+00              f04   00000000      0.000000e+00      0.000000e+00
 f05   00000000      0.000000e+00                                f05   00000000      0.000000e+00
 f06   00000000      0.000000e+00      0.000000e+00              f06   00000000      0.000000e+00      0.000000e+00
 f07   00000000      0.000000e+00                                f07   00000000      0.000000e+00
```

```
  f08   BEEE24F7     -4.651258e-01     -1.437397e-05          f08   00000000     0.000000e+00     0.000000e+00
  f09   BD89CB9C     -6.728289e-02                            f09   00000000     0.000000e+00
  f10   BF9097EC     -1.129636e+00     -1.620454e-02          f10   00000000     0.000000e+00     0.000000e+00
  f11   3EFFFFF8      4.999998e-01                            f11   00000000     0.000000e+00
  f12   40980809      4.750981e+00      1.538010e+03          f12   00000000     0.000000e+00     0.000000e+00
  f13   C01097EE     -2.259273e+00                            f13   00000000     0.000000e+00
  f14   3ECBB210      3.978429e-01      3.301569e-06          f14   00000000     0.000000e+00     0.000000e+00
  f15   BF9097E9     -1.129636e+00                            f15   00000000     0.000000e+00
  f16   BF3B56C8     -7.317929e-01     -4.171599e-04          f16   00000000     0.000000e+00     0.000000e+00
  f17   00000000      0.000000e+00                            f17   00000000     0.000000e+00
  f18   3D1E0000      3.857422e-02      2.664535e-14          f18   00000000     0.000000e+00     0.000000e+00
  f19   00000000      0.000000e+00                            f19   00000000     0.000000e+00
  f20   3FD99999      1.700000e+00      4.000000e-01          f20   00000000     0.000000e+00     0.000000e+00
  f21   9997FA04     -1.571401e-23                            f21   00000000     0.000000e+00
  f22   00000000      0.000000e+00      0.000000e+00          f22   00000000     0.000000e+00     0.000000e+00
  f23   00000000      0.000000e+00                            f23   00000000     0.000000e+00
  f24   00000000      0.000000e+00      0.000000e+00          f24   00000000     0.000000e+00     0.000000e+00
  f25   00000000      0.000000e+00                            f25   00000000     0.000000e+00
  f26   00000000      0.000000e+00      0.000000e+00          f26   00000000     0.000000e+00     0.000000e+00
  f27   00000000      0.000000e+00                            f27   00000000     0.000000e+00
  f28   00000000      0.000000e+00      0.000000e+00          f28   00000000     0.000000e+00     0.000000e+00
  f29   00000000      0.000000e+00                            f29   00000000     0.000000e+00
  f30   00000000      0.000000e+00      0.000000e+00          f30   00000000     0.000000e+00     0.000000e+00
  f31   00000000      0.000000e+00                            f31   00000000     0.000000e+00
```

■ `run/r` and `stop` : The run command is to run or continue execution. This command is different from `TSIM` run/go/cont commands. The stop command is to stop execution. `laysim-GR712RC` supports the power-down mode of GR712RC only on run mode, so stop command will break the power-down mode of the processor.

Figure 3-25. run/stop command

```
 laysim for GR712RC Dual Core Processor v3.1 by layright

 laysim>
 CORE#0 RAM file is loaded - /home/layright/laysim-gr712rc/BCC-2.2.1/dhry.elf
    - .text, addr: 0x40000000, size 60160 bytes
    - .rodata, addr: 0x4000EB00, size 2048 bytes
    - .data, addr: 0x4000F300, size 1568 bytes
    - .bss, addr: 0x4000F920, size 10688 bytes (not loaded, just info.)
    - found DWARF debug information
 Only GR712RC CORE#0 RAM file is loaded, starts from RAM
 Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

 laysim> run

 laysim> stop
 laysim-gr712rc emulator is stopped by user
```

■ `step/s` and `next/n` : The step and next commands are to single step of execution based on selected core as shown in Figure 3-26. The step command executes and disassembles one instruction, whereas the next command executes and disassembles one instruction with showing current status of the GR712RC IU registers. Theses commands are only effective in stop mode. The step over command is not supported in the control console, it is only available in the source code window. `laysim-GR712RC` supports the power-down mode of GR712RC only on run mode, so step/next/trace command would not enter the power-down mode of the processor.

Figure 3-26. step & next command

```
 laysim> core 2
 GR712RC all COREs are selected for debugging

 laysim> s
 73286288 (73286303) -E CORE#0   40009248 : 02800035   be      0x4000931c
 73286288 (73286289) -E CORE#1   40011958 : 82006001   add     %g1, 0x1, %g1

 laysim> n
 73286290 (73286303) -E CORE#0   40009248 : 02800035   be      0x4000931c
 73286290 (73286290) -E CORE#1   4001195C : 01000000   nop


  ---------------------- CORE#0 ----------------------          ---------------------- CORE#1 ----------------------
         INS       LOCALS       OUTS       GLOBALS                INS       LOCALS       OUTS       GLOBALS
   0:  400D3CC0    4006C5D0    400D3DCC    00000000          0:  4006F348    4006F370    400D26D0    00000000
   1:  F34000E2    400D3DD8    4006DB28    0000CDE9          1:  00000007    400DC334    400DC334    0000CABC
```

```
  2:  400D3DCC    400D3DC8    00000000    0000CDE8          2:  400D3EC0    00000000    4006F2C0    0000CABB
  3:  400D3CC0    400D3DD4    00000005    400A8D78          3:  4006F0E8    4006F124    4006F104    00000000
  4:  00000000    400583A0    400D26E0    00000000          4:  400168D8    4006F090    4006F208    33333333
  5:  400D3DD8    4005839C    4006F348    00000000          5:  4006F080    00000000    00000000    00000000
  6:  400DA2B0    00000007    400DA230    400D3CC0          6:  400DC3E8    4006F0A8    400DC2C0    400D3EC0
  7:  40006F74    00000000    40009138    00000000          7:  40001330    4006F2C0    40007398    00000000

 psr:F3000FE2 wim:00000020 tbr:40000890 y:00001674        psr:F3400FE7 wim:00000002 tbr:40000890 y:00000000
 [n:0 z:0 v:0 c:0 ef:0 pil:15 s:1 ps:1 et:1 cwp:2]         [n:0 z:1 v:0 c:0 ef:0 pil:15 s:1 ps:1 et:1 cwp:7]

 pc:   40009248 : 02800035   be      0x4000931c           pc:   40011960 : C3E20142   casa    [%o0] 0xa, %g2, %g1
 npc:  4000924C : 01000000   nop                          npc:  40011964 : 80A04002   subcc   %g1, %g2, %g0
```

■ `trace/t` : The trace command executes and disassembles multiple instructions as indicated count based on selected core as shown in Figure 3-27. This command is effective only in stop mode.

Figure 3-27. trace command

```
laysim> t 10

81906756 (81906756) -E CORE#0   40011950 : C2020000   ld      [%o0], %g1   ! _SMP_lock_Acquire
81906757 (81906757) -E CORE#0   40011954 : 84100001   or      %g0, %g1, %g2
81906759 (81906759) -E CORE#0   40011958 : 82006001   add     %g1, 0x1, %g1
81906760 (81906760) -E CORE#0   4001195C : 01000000   nop
81906761 (81906761) -E CORE#0   40011960 : C3E20142   casa    [%o0] 0xa, %g2, %g1
81906766 (81906766) -E CORE#0   40011964 : 80A04002   subcc   %g1, %g2, %g0
81906767 (81906767) -E CORE#0   40011968 : 32BFFFFC   bne,a   0x40011958
81906771 (81906771) -E CORE#0   40011970 : 90022004   add     %o0, 0x4, %o0
81906772 (81906772) -E CORE#0   40011974 : C2020000   ld      [%o0], %g1
81906773 (81906773) -E CORE#0   40011978 : 80A08001   subcc   %g2, %g1, %g0
```

■ `break/b` and `del/d` : The break command adds a breakpoint at [addr]/[sym] with supporting virtual address based on selected core as shown in Figure 3-28. The status of breakpoints can be checked in the breakpoint window. The del command deletes a breakpoint with [addr]/[sym].

Figure 3-28. break command

```
laysim> break sin
[CORE#0] Set breakpoint at sin(0x40005060)

laysim> run

########################################
Single Precision C/C++ Whetstone Benchmark

Calibrate
[CORE#0] Breakpoint hit at 0x40005060(sin)

laysim> reg

 ---------------------- CORE#0 --------------------
         INS       LOCALS       OUTS       GLOBALS
  0:  00000001    40016800    3FDFFFFF    00000000
  1:  00000001    40016A2C    00000000    40015400
  2:  40015400    00000C7E    00000002    3FDFFFFF
  3:  40016800    40015400    40016990    00000000
  4:  40016800    3FDFFFFF    00000001    20706F69
  5:  40015784    00000000    69786564    00000000
  6:  407FFAB0    00000000    407FFA00    00000000
  7:  400046FC    00000000    40001D80    00000000

 psr:F30010E6 wim:00000001 tbr:40000060 y:00000000
 [n:0 z:0 v:0 c:0 ef:1 pil:0 s:1 ps:1 et:1 cwp:6]

 pc:    40005060 : 9DE3BF88   save    %sp, -120, %sp
 npc:   40005064 : 03200000   sethi   %hi(0x80000000), %g1

laysim> b
[CORE#0] Breakpoint at 0x40005060

laysim> del sin
[CORE#0] Delete the breakpoint at sin(0x40005060)
```

■ `watch` and `dwatch/dw` : The watch command adds a watchpoint at [addr]/[sym] with supporting virtual address based on selected core as

shown in Figure 3-29. The status of watchpoints can be checked in the watchpoint window in GUI. The dwatch command deletes a watchpoint with [addr]/[sym]. `laysim-gr712rc` supports both read and write watchpoints, whereas `laysim-gr712rc-dbt` supports only write watchpoints

Figure 3-29. watch command

```
laysim> core 2
GR712RC all COREs are selected for debugging

laysim> watch 0x80000100
[CORE#0] Set watchpoint at 0x80000100
[CORE#1] Set watchpoint at 0x80000100

laysim> run
[CORE#1] Watchpoint hit for write at 0x80000100, new data : 0x0000000D

laysim> reg

 --------------------- CORE#0 ---------------------          --------------------- CORE#1 ---------------------
         INS       LOCALS       OUTS       GLOBALS                INS       LOCALS       OUTS       GLOBALS
  0:  40026FC8    4001BE8C    00000000    00000000           0:  0000000A    00000000    80000100    00000000
  1:  4000DE30    4001DD60    4001DD60    40001B78           1:  00000000    00000000    0000000D    00000086
  2:  00000000    4001BE20    400250A0    09010001           2:  00000000    00000000    00000001    00000000
  3:  400250A0    400250B4    00000001    00000000           3:  00000000    00000000    00000001    0000000D
  4:  400250B8    40025000    00000000    400250AC           4:  00000000    00000000    00000000    400250AC
  5:  40026FC8    4002507C    00000000    00000000           5:  40025000    00000000    00000000    00000000
  6:  4001DD78    400257C0    4001DD18    40028900           6:  4002A700    00000000    4002A6A0    40028B00
  7:  40009EAC    4001BD58    400198B4    00000000           7:  4000E4EC    00000000    40004010    00000000

 psr:F39000E3 wim:00000040 tbr:400008A0 y:FFFFFFFE          psr:F34000E7 wim:00000040 tbr:400008A0 y:00000000
 [n:1 z:0 v:0 c:1 ef:0 pil:0 s:1 ps:1 et:1 cwp:3]           [n:0 z:1 v:0 c:0 ef:0 pil:0 s:1 ps:1 et:1 cwp:7]

 pc:   40001B7C : C0838020   lda     [%sp] 0x1, %g0         pc:   4001B074 : D2220000   st      %o1, [%o0]
 npc:  40001B80 : 30BFFFFE   ba,a    0x40001b78             npc:  4001B078 : 80A2E000   subcc   %o3, 0x0, %g0
```

■ `icache` : The icache command controls GR712RC instruction cache operation. This command is only effective in stop mode. There are 6 sub-commands for the icache command. This command is only available in `laysim-gr712rc/laysimgr712rc-cli`.

Figure 3-30. icache command

```
laysim> icache 0 dump

CORE#0 icache set 0
40008000 : ff : 7fffff47 81e80000 81c7e008 91e82000 10bffff8 f2062004 4000011a 90100018
4000d020 : ff : a6102066 80a4e047 0480003c 07100057 07100057 d80fbef7 10bff66e ac10e110
40000040 : ff : 91d02000 01000000 01000000 01000000 a1480000 2910001d 81c520ec 01000000
40002060 : ff : 91a0192a d13fbfe0 40000bc1 d01fbfe0 d307bff8 d107bff0 9da208a9 95a00020
40007080 : ff : d43ba048 91a38948 e51ba048 03100056 e1186198 a1a48950 03100057 c1186070
4000e0a0 : ff : c027bfd8 ac05bfff 80a5e000 068001a1 a6102000 80a5a000 16800182 e2062024
4000e0c0 : ff : a624c016 a0200016 a8100016 80a6e009 18800181 ac102000 80a6e005 04800004
400020e0 : ff : 40001647 01000000 d027bff0 01000000 80a22000 d024210c d107bff0 16800005
40014100 : ff : 92102000 81c7e008 81e80000 c0262008 c026200c 80a2a024 12bffff4 82062010
...
...
```

■ `dcache` : The dcache command controls GR712RC data cache operation. This command is only effective in stop mode. There are 4 sub-commands for the dcache command. This command is only available in `laysim-gr712rc/laysimgr712rc-cli`.

Figure 3-31. dcache command

```
laysim> dcache 0 dump

CORE#0 icache set 0
40004000 : ff : 80a62000 d027beb8 02800006 0310003b c2062018 80a06000 0280016f 0310003b
40001020 : ff : 80a30000 12800009 1110003e 90122120 15100048 9412a2e0 96228008 9332e003
40001040 : ff : 400002f4 01000000 2310003e e0246120 400002fb 01000000 400002c6 01000000
40000060 : ff : a1480000 29100006 81c52258 01000000 91d02000 01000000 01000000 01000000
40001080 : ff : 1110003a 901222e0 400003b9 01000000 4000368c 01000000 400002b6 01000000
400010a0 : ff : 21100048 d0042170 2310003c d2046314 400000c6 94100000 400003b4 01000000
400090c0 : ff : 900220f0 0310003c c2006338 d0006020 80a22000 22800004 1110003d 81c3e008
400040e0 : ff : c027bea4 01000000 ba07bf58 c027bea0 01000000 a810001a c027be9c c027be98
40001100 : ff : 81c7e008 81e80000 9de3bfa0 0310003e 1110003e 92106120 90122120 92224008
```

```
...
...
```

■ `core` : The core command shows the status of each core and selects the core for debugging as shown in Figure 3-32. The selected core for debugging is used for all operation.

Figure 3-32. core command

```
laysim> core 2
GR712RC all COREs are selected for debugging

laysim> core
GR712RC CORE#0 is execution mode
GR712RC CORE#1 is execution mode
GR712RC all COREs are selected for debugging

laysim> reg

-------------------- CORE#0 --------------------         -------------------- CORE#1 --------------------
        INS       LOCALS       OUTS       GLOBALS              INS       LOCALS       OUTS       GLOBALS
  0:  40026FC8    4001BE8C    00000000    00000000          0:  0000000A    00000000    80000100    00000000
  1:  4000DE30    4001DD60    4001DD60    40001B78          1:  00000000    00000000    0000000A    00000084
  2:  00000000    4001BE20    400250A0    09010001          2:  00000000    00000000    00000001    00000001
  3:  400250A0    400250B4    00000001    00000000          3:  00000000    00000000    00000001    0000000D
  4:  400250B8    40025000    00000000    400250AC          4:  00000000    00000000    00000000    400250AC
  5:  40026FC8    4002507C    00000000    00000000          5:  40025000    00000000    00000000    00000000
  6:  4001DD78    400257C0    4001DD18    40028900          6:  4002A700    00000000    4002A6A0    40028B00
  7:  40009EAC    4001BD58    400198B4    00000000          7:  4000E4EC    00000000    4000401C    00000000

 psr:F39000E3 wim:00000040 tbr:400008A0 y:FFFFFFFE         psr:F30000E7 wim:00000040 tbr:400008A0 y:00000000
 [n:1 z:0 v:0 c:1 ef:0 pil:0 s:1 ps:1 et:1 cwp:3]          [n:0 z:0 v:0 c:0 ef:0 pil:0 s:1 ps:1 et:1 cwp:7]

 pc:   40001B7C : C0838020   lda     [%sp] 0x1, %g0        pc:   4001B068 : C6220000   st      %g3, [%o0]
 npc:  40001B80 : 30BFFFFE   ba,a    0x40001b78            npc:  4001B06C : 10BFFFF7   ba      0x4001b048
```

■ `hitstory/hist` : The history [core] command shows the maximum 1000 executed instructions of selected core as shown in Figure. 3-33. This command is only available in `laysim-gr712rc/laysimgr712rc-cli`.

Figure 3-33. history command

```
laysim> hist 0

81903439 -E CORE#0   400090FC : 01000000   nop
81903440 -E CORE#0   40009100 : 82042064   add     %l0, 0x64, %g1
81903441 -E CORE#0   40009104 : C227BFE4   st      %g1, [%fp - 28]
81903445 -E CORE#0   40009108 : 01000000   nop
81903446 -E CORE#0   4000910C : 82042068   add     %l0, 0x68, %g1
81903447 -E CORE#0   40009110 : C227BFF0   st      %g1, [%fp - 16]
81903451 -E CORE#0   40009114 : 2B100160   sethi   %hi(0x40058000), %l5
81903452 -E CORE#0   40009118 : AA15639C   or      %l5, 0x39c, %l5
81903453 -E CORE#0   4000911C : A8056004   add     %l5, 0x4, %l4
81903454 -E CORE#0   40009120 : C4042080   ld      [%l0 + 0x80], %g2
81903455 -E CORE#0   40009124 : 80A0A000   subcc   %g2, 0x0, %g0
81903457 -E CORE#0   40009128 : 128000C1   bne     0x4000942c
81903460 -E CORE#0   4000912C : 8088A001   andcc   %g2, 0x1, %g0
81903461 -E CORE#0   40009130 : A4062108   add     %i0, 0x108, %l2
81903462 -E CORE#0   40009134 : 9207BFF8   add     %fp, -8, %o1
81903463 -E CORE#0   40009138 : 40002206   call    0x40011950
81903464 -E CORE#0   4000913C : 90100012   or      %g0, %l2, %o0
81903465 -E CORE#0   40011950 : C2020000   ld      [%o0], %g1   ! _SMP_lock_Acquire
81903466 -E CORE#0   40011954 : 84100001   or      %g0, %g1, %g2
81903468 -E CORE#0   40011958 : 82006001   add     %g1, 0x1, %g1
81903469 -E CORE#0   4001195C : 01000000   nop
81903470 -E CORE#0   40011960 : C3E20142   casa    [%o0] 0xa, %g2, %g1
81903475 -E CORE#0   40011964 : 80A04002   subcc   %g1, %g2, %g0
81903476 -E CORE#0   40011968 : 32BFFFFC   bne,a   0x40011958
81903480 -E CORE#0   40011970 : 90022004   add     %o0, 0x4, %o0
81903481 -E CORE#0   40011974 : C2020000   ld      [%o0], %g1
81903482 -E CORE#0   40011978 : 80A08001   subcc   %g2, %g1, %g0
81903484 -E CORE#0   4001197C : 12BFFFFE   bne     0x40011974
...
...
```

■ `perf` : The perf command shows the performance statistics of `laysim-GR712RC` as shown in Figure 3-34. This command is only effective in stop mode.

Figure 3-34. perf command

```
laysim> perf

 Performance statistics for CORE#0
   Cycles       : 64222527
   Instructions : 215588
   Overall CPI  : 5.09
   CORE#0 performance (100MHz) : 19.65 MOPS (19.65 MIPS, 0.00 MFLOPS)
   Simulated time              : 642.23 ms
   Processor utilization       : 1.71 %

 Performance statistics for CORE#1
   Cycles       : 64222487
   Instructions : 32157740
   Overall CPI  : 2.00
   CORE#1 performance (100MHz) : 50.07 MOPS (50.07 MIPS, 0.00 MFLOPS)
   Simulated time              : 642.22 ms
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 114.13 %
   laysim-gr712rc performance  : 57.53 MIPS
   Wall clock                  : 0.56 s
```

■ `fastuart` : The fastuart command controls the baudrate of APBUARTs. When fastuart is enabled, the baudrate of APBUART works as infinite.

■ `mmu` : The mmu [core] command shows the current status of GR712RC MMU.

Figure 3-35. mmu command

```
laysim> mmu 0
CORE#0 - MMU: 0x00920001, ctx: 0x40, ctxptr: 0x408C1000, fsr: 0x00000384, far: 0x5033D000

laysim> mmu 1
CORE#1 - MMU: 0x00920001, ctx: 0x3F, ctxptr: 0x408C1000, fsr: 0x00000344, far: 0x5026F000
```

■ `walk` : The walk [core] [vaddr] command shows a MMU table walk for the given virtual address as shown in Figure 3-36.

Figure 3-36. walk command

```
laysim> walk 0 0xF000BA98
CORE#0 - vaddr : 0xF000BA98 (Index1: 0xF0, Index2: 0x00, Index3: 0x0B)
       - Context    70 0x408C1118 PTD: 0x0408CF01
       - Region    240 0x408CF3C0 PTE: 0x0400009E
       --> physical address : 0x4000BA98

laysim> walk 1 0xF0012C70
CORE#1 - vaddr : 0xF0012C70 (Index1: 0xF0, Index2: 0x00, Index3: 0x12)
       - Context    66 0x408C1108 PTD: 0x0408CFC1
       - Region    240 0x408CFFC0 PTE: 0x0400009E
       --> physical address : 0x40012C70
```

■ `vdis` : The vdis command disassembles [count] instructions at [vaddr] with [count] as shown in Figure 3-37.

Figure 3-37. vdis command

```
laysim> vdis 0 0xF000BA98
4000BA98 -> F000BA98 : EA800320   lda     [%g0] 0x19, %l5
4000BA9C -> F000BA9C : AA156002   or      %l5, 0x2, %l5
4000BAA0 -> F000BAA0 : EAA00320   sta     %l5, [%g0] 0x19
4000BAA4 -> F000BAA4 : E01BA000   ldd     [%sp + 0x0], %l0
4000BAA8 -> F000BAA8 : E41BA008   ldd     [%sp + 0x8], %l2
4000BAAC -> F000BAAC : E81BA010   ldd     [%sp + 0x10], %l4
4000BAB0 -> F000BAB0 : EC1BA018   ldd     [%sp + 0x18], %l6
4000BAB4 -> F000BAB4 : F01BA020   ldd     [%sp + 0x20], %i0
4000BAB8 -> F000BAB8 : F41BA028   ldd     [%sp + 0x28], %i2
4000BABC -> F000BABC : F81BA030   ldd     [%sp + 0x30], %i4

laysim> vdis 1 0xF0012C70
40012C70 -> F0012C70 : 81C7E008   jmpl    %i7 + 0x8, %g0
40012C74 -> F0012C74 : 81E80000   restore
```

```
40012C78 -> F0012C78 : 9DE3BFA0   save    %sp, -96, %sp
40012C7C -> F0012C7C : 033C2178   sethi   %hi(0xf085e000), %g1
40012C80 -> F0012C80 : 7FFFE62A   call    0xf000c528
40012C84 -> F0012C84 : FA00611C   ld      [%g1 + 0x11c], %i5
40012C88 -> F0012C88 : 01000000   nop
40012C8C -> F0012C8C : 01000000   nop
40012C90 -> F0012C90 : 01000000   nop
40012C94 -> F0012C94 : 01000000   nop
```

■ `rtems` : The rtems command shows the `RTEMS` information based on `RTEMS` awareness of `laysim-GR712RC`.

Figure 3-38. rtems command

```
laysim> rtems
Detect RTEMS v5.1(SMP) with thread information

laysim> rtems task

+--------+-------+----------+------------+------------+-----+--------------------+----------------------------------------+-------------------------+
|  Name  | CORE# | Type     | ID         | TCB        | Pri | Time (h:m:s)       | Entry point                            | State                   |
+--------+-------+----------+------------+------------+-----+--------------------+----------------------------------------+-------------------------+
|  IDLE  | core0 | internal | 0x09010001 | 0x40030028 | 255 | 0:00:01.0365209658 | (0x40001D30) leon3_power_down_loop     | READY                   |
|  IDLE  | core1 | internal | 0x09010002 | 0x40030518 | 255 | 0:00:10.4084105875 | (0x40001D30) leon3_power_down_loop     | READY                   |
|  TA01  | core1 | classic  | 0x0A010002 | 0x4002C330 |   2 | 0:00:00.0264209209 | (0x4000163C) Test_task                 | WAITING_FOR_TIME        |
|  TA02  | core1 | classic  | 0x0A010003 | 0x4002C820 |   2 | 0:00:00.0915115797 | (0x4000163C) Test_task                 | WAITING_FOR_TIME        |
|  TA03  | core0 | classic  | 0x0A010004 | 0x4002CD10 |   2 | 0:00:00.0088746909 | (0x4000163C) Test_task                 | WAITING_FOR_TIME        |
|  TA04  | core0 | classic  | 0x0A010005 | 0x4002D200 |   2 | 0:00:00.0088798449 | (0x4000163C) Test_task                 | WAITING_FOR_TIME        |
|  TA05  | core0 | classic  | 0x0A010006 | 0x4002D6F0 |   2 | 0:00:00.0088794154 | (0x4000163C) Test_task                 | WAITING_FOR_TIME        |
|  TA06  | core0 | classic  | 0x0A010007 | 0x4002DBE0 |   2 | 0:00:00.0088798449 | (0x4000163C) Test_task                 | WAITING_FOR_TIME        |
+--------+-------+----------+------------+------------+-----+--------------------+----------------------------------------+-------------------------+

laysim> rtems core 0

RTEMS5.1 is configured as MAX 4 cores

 ############ RTEMS5 CORE#0 Information ############
 - cpu_per_cpu.fsr               : 0x00000000
 - interrupt_stack_low           : 0x40027100
 - interrupt_stack_high          : 0x40028100
 - isr_nest_level                : 0x00000000
 - isr_dispatch_disable          : 0x00000000
 - thread_dispatch_disable_level : 0x00000000
 - dispatch_necessary            : 0
 - executing                     : 0x40030028 (_Thread_Objects)
 - heir                          : 0x40030028 (_Thread_Objects)
 - Interrupt_frame
   - Stack_frame.l0              : 0x00000000
   - Stack_frame.l1              : 0x00000000
   - Stack_frame.l2              : 0x00000000
   - Stack_frame.l3              : 0x00000000
   - Stack_frame.l4              : 0x00000000
   - Stack_frame.l5              : 0x00000000
   - Stack_frame.l6              : 0x00000000
   - Stack_frame.l7              : 0x00000000
   - Stack_frame.i0              : 0x00000000
   - Stack_frame.i1              : 0x00000000
   - Stack_frame.i2              : 0x00000000
   - Stack_frame.i3              : 0x00000000
   - Stack_frame.i4              : 0x00000000
   - Stack_frame.i5              : 0x00000000
   - Stack_frame.i6/fp           : 0x00000000
   - Stack_frame.i7              : 0x00000000
   - Stack_frame.return_address  : 0x00000000
   - Stack_frame.saved_arg0      : 0x00000000
   - Stack_frame.saved_arg1      : 0x00000000
   - Stack_frame.saved_arg2      : 0x00000000
   - Stack_frame.saved_arg3      : 0x00000000
   - Stack_frame.saved_arg4      : 0x00000000
   - Stack_frame.saved_arg5      : 0x00000000
   - Stack_frame.pad0            : 0x00000000
   - psr                         : 0xF30000C4
   - pc                          : 0x400180FC
   - npc                         : 0x40018100
   - g1                          : 0xF30000E5
   - g2                          : 0x00000006
   - g3                          : 0x00000005
   - g4                          : 0x00000040
   - g5                          : 0x00000080
   - reserved                    : 0x00000000
   - g7                          : 0x00000000
   - i0                          : 0x40027F70
   - i1                          : 0x400301C0
   - i2                          : 0xF3401FE5
   - i3                          : 0x400317C0
   - i4                          : 0x00000000
   - i5                          : 0x00000002
```

```
    - i6/fp                     : 0x40031A98
    - i7                        : 0x4001A44C
    - y                         : 0xFFFFFFFE
    - tpc                       : 0x40030758
  - cpu_usage_timestamp         : 0x00000001303393AA (0:00:01.808686506)
  - Watchdog
    - Lock.next_ticket          : 0x0000053A
    - Lock.now_serving          : 0x0000053A
    - ticks                     : 0x0000000000000532
    - Header[0].Watchdogs       : 0x4002D808
    - Header[0].first           : 0x4002CE28
    - Header[1].Watchdogs       : 0x00000000
    - Header[1].first           : 0x00000000
    - Header[2].Watchdogs       : 0x00000000
    - Header[2].first           : 0x00000000
  - Lock.next_ticket            : 0x0000001A
  - Lock.now_serving            : 0x0000001A
  - Lock_context                : 0x00000000
  - Threads_in_need_for_help
    - Head.Node.next            : 0x40031AD8
    - Head.Node.previous        : 0x00000000
    - Head.fill                 : 0x40031AD4
  - message                     : 0x00000000
  - Scheduler.control           : 0x40021C60 (_Scheduler_Table)
  - Scheduler.context           : 0x40031490 (_Configuration_Scheduler_EDF_SMP_dflt)
  - Scheduler.idle_if_          : 0x00000000
  - ancestor                    : 0x40030028
  - data                        : 0x00000000
  - state                       : 0x00000003 (PER_CPU_STATE_UP)
  - Jobs
    - Lock.next_ticket          : 0x000001D1
    - Lock.now_serving          : 0x000001D1
    - head                      : 0x00000000
    - tail                      : 0x40027E10
  - online/boot                 : 0x0101
  - record                      : 0x00000000
```

■ `info sys` : The info sys command shows the GR712RC system information as shown in Figure 3-39.

■ `info reg` : The info reg command shows all the registers of [core_name] as shown in Figure 3-39.

Figure 3-39. info sys & reg command

```
laysim> info sys

  Address      Description                            Core name
  ----------   --------------------                   ---------
  0x80000000   Fault Tolerant Memory Controller       ftmctrl
  0x80000100   UART Serial Interface #0                apbuart0
  0x80000200   Multiprocessor Interrupt Controller     irqmp
  0x80000300   General Purpose Timer Unit              gptimer
  0x80000400   SPI Controller                         spictrl
  0x80000500   CAN Bus Multiplexer (Dummy)
  0x80000600   General Purpose Register (Dummy)
  0x80000700   ASCS Controller (Dummy)
  0x80000800   SLINK Serial Bus (Dummy)
  0x80000900   General Purpose I/O Port #0            grgpio0
  0x80000A00   General Purpose I/O Port #1            grgpio1
  0x80000B00   GRTM - CCSDS Telemetry Encoder (Dummy)
  0x80000C00   I2C Master                             i2cmst
  0x80000D00   Clock Gating Unit                      grclkgate
  0x80000E00   10/100 Ethernet MAC                    greth
  0x80000F00   AHB Status Register                    ahbstat

  0x80100000   FTAHBRAM Register                      ftahbram
  0x80100100   UART Serial Interface #1               apbuart1
  0x80100200   UART Serial Interface #2               apbuart2
  0x80100300   UART Serial Interface #3               apbuart3
  0x80100400   UART Serial Interface #4               apbuart4
  0x80100500   UART Serial Interface #5               apbuart5
  0x80100600   GPTIMER with Time Latch Capability     grtimer
  0x80100800   GRSPW2 Spacewire Link #0               grspw0
  0x80100900   GRSPW2 Spacewire Link #1               grspw1
  0x80100A00   GRSPW2 Spacewire Link #2               grspw2
  0x80100B00   GRSPW2 Spacewire Link #3               grspw3
  0x80100C00   GRSPW2 Spacewire Link #4               grspw4
  0x80100D00   GRSPW2 Spacewire Link #5               grspw5

  0xFFF00000   Core 1553BRM                           1553brm
  0xFFF10000   GRTC - Telecommand Decoder (Dummy)
  0xFFF20000   SATCAN (Dummy)
  0xFFF30000   CAM-2.0 Interface #0                   canoc0
  0xFFF30100   CAM-2.0 Interface #1                   canoc1
```

```
laysim> info reg gptimer

  General Purpose Timer Unit
      0x80000300  Scaler Value Register               0x00000029
      0x80000304  Scaler Reload Value Register        0x00000063
      0x80000308  Configuration Register              0x00000144
      0x80000310  Timer 0 Counter Value Register      0x0000198F
      0x80000314  Timer 0 Reload Value Register       0x0000270F
      0x80000318  Timer 0 Control Register           0x0000001F
      0x80000320  Timer 1 Counter Value Register      0xFF34DF11
      0x80000324  Timer 1 Reload Value Register       0xFFFFFFFF
      0x80000328  Timer 1 Control Register           0xFF34DF11
      0x80000330  Timer 2 Counter Value Register      0xFFFFFFFF
      0x80000334  Timer 2 Reload Value Register       0xFFFFFFFF
      0x80000338  Timer 2 Control Register           0xFFFFFFFF
      0x80000340  Timer 3 Counter Value Register      0xFFFFFFFF
      0x80000344  Timer 3 Reload Value Register       0xFFFFFFFF
      0x80000348  Timer 3 Control Register           0xFFFFFFFF
```

■ `event` : The event command shows the pending events in the event queue of `laysim-GR712RC`.

■ `memdump` : Reads [wcnt] size from memory [addr]/[sym] and stores it into [file] in binary format. Note that the data is stored in little-endian format.

Figure 3-40. memdump command

```
laysim> mem Init
[CORE#0] Address of Init is 0x4000130C
 40001300 : 90122240 81C7E008 81E80000 9DE3BF78     .."@...........x
 40001310 : 92102000 11100086 4000670E 901223F0     .. .....@.g...#.
 40001320 : 4000014D 01000000 821027C4 C227BFE4     @..M......'..'..

laysim> memdump Init 10 output.bin
RAM from 0x4000130C was saved to : output.bin
```

■ `rundelta` : This executes for [dcycles] based on the current cycle, and is only available in `laysim-gr712rc-cli`.

Figure 3-41. rundelta command

```
laysim> s
102753530 (102753534) -E CORE#0   400012F4 : 91A24828   fadds   %f9, %f8, %f8

laysim> rundelta 1000
Current cycles : 102753538
Run till 102754538 cycles

laysim> s
102754540 (102754540) -E CORE#0   40001308 : 95A208A9   fsubs   %f8, %f9, %f10
```

■ `reset` : This command resets all state information of the `laysim-GR712RC`.

■ `load` : This loads a GR712RC RAM executable in the CLI version. To load a new RAM executable, `laysim-GR712RC` must first be initialized using reset command. In the GUI version, executing the reset command activates the File Open toolbar again, allowing a new executable to be loaded.

Figure 3-42. load command

```
[layright@CentOS BCC-2.2.1]$ laysim-gr712rc-cli -core0 dhry.elf
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - dhry.elf
   - .text, addr: 0x40000000, size 60160 bytes
   - .rodata, addr: 0x4000EB00, size 2048 bytes
   - .data, addr: 0x4000F300, size 1568 bytes
```

```
   - .bss, addr: 0x4000F920, size 10688 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

laysim> run
Execution starts, 1000000 runs through Dhrystone

laysim> reset
Reset laysim-GR712RC state

laysim> load stanford.elf
CORE#0 RAM file is loaded - stanford.elf
   - .text, addr: 0x40000000, size 78784 bytes
   - .rodata, addr: 0x400133C0, size 2288 bytes
   - .data, addr: 0x40013CB0, size 1568 bytes
   - .bss, addr: 0x400142D0, size 95040 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

laysim> run
Starting
    Perm  Towers  Queens   Intmm      Mm  Puzzle   Quick  Bubble    Tree     FFT
  402700  238500  212850  202400  167684 1283700  206550  296917  339317  264667

Nonfloating point composite is     432451

Floating point composite is
CORE#0 halts because of Error Mode [28854806cycles/0.242754sec]
598927
laysim>
```

■ `batch` : Reads the batch file [file] and executes it automatically. This command is only available in the CLI version.

Figure 3-43. batch command

```
[layright@CentOS RCC-1.3.1]$ laysim-gr712rc-dbt-cli
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
--> No CORE#0 RAM file is loaded

laysim> batch gr712rc.bat
Read batch file from gr712rc.bat

CORE#0 RAM file is loaded - rtems-hello.elf
   - .text, addr: 0x40000000, size 170800 bytes
   - .rtemsroset, addr: 0x40029B30, size 160 bytes
   - .rtemsstack, addr: 0x40029C00, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x4002BC00, size 4720 bytes
   - .bss, addr: 0x4002CE80, size 15792 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
[CORE#0] Set breakpoint at Init(0x400012CC)

       0 (0) E CORE#0   40000000 : A0100000   or      %g0, %g0, %l0   ! start
       1 (1) E CORE#0   40000004 : 29100004   sethi   %hi(0x40001000), %l4
       2 (2) E CORE#0   40000008 : 81C52000   jmpl    %l4, %g0
       4 (4) E CORE#0   4000000C : A6102000   or      %g0, 0x0, %l3
       5 (5) E CORE#0   40001000 : 03100000   sethi   %hi(0x40000000), %g1   ! hard_reset
       6 (6) E CORE#0   40001004 : 82106000   or      %g1, 0x0, %g1
       7 (7) E CORE#0   40001008 : 81980001   wr      %g0, %g1, %tbr
       8 (8) E CORE#0   4000100C : 83480000   rd      %psr, %g1
       9 (9) E CORE#0   40001010 : 84006001   add     %g1, 0x1, %g2
      10 (10) E CORE#0   40001014 : 8408A007   and     %g2, 0x7, %g2
[CORE#0] Breakpoint hit at 0x400012CC(Init)


 -------------------- CORE#0 --------------------
         INS       LOCALS       OUTS       GLOBALS
  0:  4002D040    40031B58    400301A0    00000000
  1:  40013B18    40027ACC    40031B58    400012CC
  2:  00000000    4002C958    4002C944    0A010001
  3:  4002C944    4002C800    00000001    0A010001
  4:  4002D040    4002C95C    4002B910    40031B5C
  5:  40027B14    4002C920    400278F0    00000000
  6:  40031B70    4002D310    40031B10    4002FC00
  7:  4000F2D0    4002D09C    40010778    00000000

 psr:F30000E3 wim:00000040 tbr:400008A0 y:FFFFFFFE
 [n:0 z:0 v:0 c:0 ef:0 pil:0 s:1 ps:1 et:1 cwp:3]

 pc:    400012CC : 9DE3BFA0   save    %sp, -96, %sp
 npc:   400012D0 : 1110009E   sethi   %hi(0x40027800), %o0

 Performance statistics for CORE#0
```

```
    Cycles       : 545727
    Instructions : 345493
    CORE#0 performance (100MHz) : 63.31 MIPS
    Simulated time              : 5.46 ms

  Performance of laysim-gr712rc
    Real-time performance       : 65.51 %
    laysim-gr712rc performance  : 41.48 MIPS
    Wall clock                  : 0.01 s
[layright@CentOS RCC-1.3.1]$
```

- `save` : This command stores the state information of `laysim-GR712RC`, including loadable modules, into [file]. This command is supported in both CLI and GUI versions. The save() callback function of each loadable module is invoked to store its state information. Note that saved files are not compatible across different emulators.

- `restore` : This command restores the previously saved state of `laysim-GR712RC` from [file]. If loadable modules are used, their states are restored as well. This command is supported in both CLI and GUI versions. Note that saved files are not compatible across different emulators.

Figure 3-44. save & restore command

```
[layright@CentOS RCC-1.3.1]$ laysim-gr712rc-cli -core0 rtems-rmap.elf -mod ../Loadable/loadable-SpW-Nodes/laysim_user_spw_node_module1.so
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

Loaded user loadable module ../Loadable/loadable-SpW-Nodes/laysim_user_spw_node_module1.so

CORE#0 RAM file is loaded - rtems-rmap.elf
   - .text, addr: 0x40000000, size 237664 bytes
   - .rtemsroset, addr: 0x4003A060, size 176 bytes
   - .rtemsstack, addr: 0x4003A140, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x4003C140, size 8288 bytes
   - .bss, addr: 0x4003E1C0, size 106640 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

laysim> run
#### To change configuration of this example, you can set a breakpoint
#### on label <rmap_conf_label> and modify the following global variables:
## remote_base_address: 0x40000000 /* Base address on Remote. This is the address to which the data is copied. */
## remote_dst_address: 0xfe /* SpW Destination address. */
## remote_dst_key: 0x00/* SpW Destination Key */
## source_src_address: 0x20 /* SpW Source address */
## rmap_dev_idx: 0 /* This parameter chooses which initiator GRSPW2 core is used */
## route_entry { nr; /* Number of addresses in dstadr array */, dstadr[16];/* Path Addresses */}
## routetab[ROUTE_TO]={2,{1,5,0, ...}
## routetab[ROUTE_FROM]={2,{1,9,0, ...}
## path_addressing: 0 /* Use path addressing (!=0) or not (0) */
## rmap_pkt_size: 992
## rmap_data_bytes: 4096 /* Data bytes to send over RMAP */
#### RTEMS RMAP example

Setting up SpaceWire router
Configuring Router
Failed to open SpW Router
Router_configure: Error -1
Failed router initialization.
Limiting to 1 SpaceWire devices
Initializing SpaceWire device 0
 After Link Start: 5

Test built for at least 1 AMBA ports.
Note that the SpW-links will not be used.
There are 6 GRSPW cores in the system.
System clock: 10000 us / tick

Starting GRSPW0: DMA Started Successfully
RMAP connection is working
Initializing data array

Starting transmission type 1: one packet at a time

RMAP write cycle completed in 0.000694 seconds
RMAP read cycle completed in 0.000725 seconds
Data bytes sent: 4096, Packet size: 992, Packet count: 5
Wrote 4096 bytes to target in 0.000694 seconds. WRITE RATE: 5763.880787 KiB/s
Read 4096 bytes from target in 0.000725 seconds. READ RATE: 5517.006248 KiB/s
Wrote 5 packets to target in 0.000694 seconds. WRITE RATE: 7204.850
```

```
laysim> save check1

 laysim-GR712RC state saved to : check1.v3.1.cli.lay
 --> Total events : 1
 0001    0x43c650   0x00000000    56484181
[Node#0] SpW Node state saved to : check1.v3.1.cli.lay-spw

laysim> quit

[layright@CentOS RCC-1.3.1]$ laysim-gr712rc-cli -mod ../Loadable/loadable-SpW-Nodes/laysim_user_spw_node_module1.so
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

Loaded user loadable module ../Loadable/loadable-SpW-Nodes/laysim_user_spw_node_module1.so
--> No CORE#0 RAM file is loaded

laysim> restore check1
 laysim-GR712RC state restored from : check1.v3.1.cli.lay
 --> Total events : 1
 0001    0x43c650   0x00000000    56484181
   - Detect RTEMS v5.1(SMP) with thread information
[Node#0] SpW Node state restored from : check1.v3.1.cli.lay-spw

laysim> run
983 pkt/s
Read 5 packets from target in 0.000725 seconds. READ RATE: 6896.257810 pkt/s


EXAMPLE 1 SUCCESSFULLY COMPLETED.


Starting transmission type 2: all packets at a time

RMAP write cycle completed in 0.000527 seconds
RMAP read cycle completed in 0.000480 seconds
Data bytes sent: 4096, Packet size: 992, Packet count: 5
Wrote 4096 bytes to target in 0.000527 seconds. WRITE RATE: 7590.641783 KiB/s
Read 4096 bytes from target in 0.000480 seconds. READ RATE: 8333.399230 KiB/s
Wrote 5 packets to target in 0.000527 seconds. WRITE RATE: 9488.302228 pkt/s
Read 5 packets from target in 0.000480 seconds. READ RATE: 10416.749038 pkt/s

EXAMPLE 2 SUCCESSFULLY COMPLETED.

CORE#0 halts because of Error Mode [80197698cycles/0.810011sec]
laysim>
```

- `regupdate` : Enable or disable the real-time update of APB registers in the GUI version of `laysim-GR712RC`.

- `fastuart` : Configure APBUART#0 to operate either in infinite mode or at the correct baudrate.

- `wake` : Forcibly wakes up the CORE#n specified by [core]. This command is used for XNG.

- `xng-setup` : Set up XNG RAM execution with XCF address.

- `gdb` : Start GDB server only for CLI version.

- `aload` : Load additional ELF file without symbol and debugging information. A newly added feature in `laysim-GR712RC` v3.1 allows simultaneous loading of both the GRBOOT ROM image and the VxWorks ROM image into MRAM, which is required to run VxWorks 7 from ROM. To do this, first the GRBOOT ROM image using the ROM Loader, and then load the VxWorks ROM image using the `aload` command. After completing these steps, VxWorks 7 can successfully boot from ROM.

Figure 3-45. aload command

```
[layright@CentOS VxWorks7]$ laysim-gr712rc-dbt-cli -rom vxWorks7-boot.rom
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
ROM file is loaded - vxWorks7-boot.rom
```

```
   - .data, addr: 0x00000000, size 11834 bytes
GR712RC ROM file is loaded, starts from ROM

laysim> aload vxWorks7-rom.img
   - .data, addr: 0x00040000, size 1404392 bytes

laysim> run

DEBUG: iu: initialized
INFO:  icache: PASS
DEBUG: iu: tested

  GRBOOT lite-1.4.2
  Copyright (C) 2023, Frontgrade Gaisler AB - all rights reserved

DEBUG: bsp_inittable0
DEBUG: bsp_inittable3
INFO:  washing... done
INFO:  memtest/addrbus: PASS
INFO:  memtest/seqdata: PASS
DEBUG: bss cleared
DEBUG: crt copied
DEBUG: fpu: init...
DEBUG: call init_main()
CFG_ROM: START=0x00000000, SIZE=0x00008000
CFG_RAM: START=0x40000000, SIZE=0x00800000
HIRAM:   START=0x406ff000, SIZE=0x00100000
REPRAM:  START=0x407ff000, SIZE=0x00001000
CFG_ASM: START=0x00040000
INFO:  romcheck: PASS
INFO:  dcache: PASS
INFO:  fpu: PASS
DEBUG: clkgate0: 80000d00 DISABLE=00000fff, ENABLE=00000000
DEBUG: irqmp0:   80000200 2 processors
DEBUG: gptimer0: 80000300 4 subtimers
DEBUG: gptimer1: 80100600 2 subtimers
precopy:  verify ASW header 00040000: PASS
precopy:    id=12340003
precopy:  verify section data in ASM
precopy:    00: flags=00000013, src=000000d2, len=00051c88, sum=8414: NOCHECK
precopy:    01: flags=00000013, src=00051d5a, len=00000077, sum=fc22: NOCHECK
precopy:    02: flags=00000013, src=00051dd1, len=000000ab, sum=485d: NOCHECK
precopy:    03: flags=00000013, src=00051e7c, len=00000035, sum=5082: NOCHECK
precopy:    04: flags=00000013, src=00051eb1, len=00003cc9, sum=c602: NOCHECK
copy:       00: dst=40003000, len=000936f7
copy:       01: dst=40250be0, len=00000081
copy:       02: dst=40250de4, len=0000016d
copy:       03: dst=402513a0, len=00000045
copy:       04: dst=40252000, len=0000b2ea
postcopy: verify section data in RAM
postcopy:   00: PASS
postcopy:   01: PASS
postcopy:   02: PASS
postcopy:   03: PASS
postcopy:   04: PASS
DEBUG: installing boot entries...
DEBUG:
CPU#1: entry: 0x40004000DEBUG: get boot entry...
INFO:  CPU0: entry: 0x40004000, stack: 0x407ff000
DEBUG: bsp_fini0_each_cpu(0)
DEBUG: gptimer_reload_watchdog
DEBUG: bsp_finitable3
DEBUG: asw_dispatch()
0x2bca58 (tRootTask): usrSmpInit
0x2bca58 (tRootTask): cpu 1 enable
0x2bca58 (tRootTask): kernelCpuEnableInternal
0x2bca58 (tRootTask): kernel cpu entry = 0x001b8e00


 ________          ________
 \77777777\        /77777777/
  \77777777\      /77777777/
   \77777777\    /77777777/
    \77777777\  /77777777/
     \77777777\ \77777/             VxWorks 7 SMP 32-bit
      \77777777\ \7777/
       \77777777\ \77/
        \77777777\ \7/     Release version: GAISLER0408-LEON-SR0650
         \77777777\ -      Build date: Oct 22 2025 19:03:08
          \77777777\
           \7777777/       Copyright Wind River Systems, Inc.
            \77777/ -                1984-2025
             \777/ /7\
              \7/ /777\
               - -------

                   Board: GR712RC-BOARD
               CPU Count: 2
          OS Memory Size: ~7MB
        ED&R Policy Mode: Permanently Deployed


 Adding 7167 symbols for standalone.
```

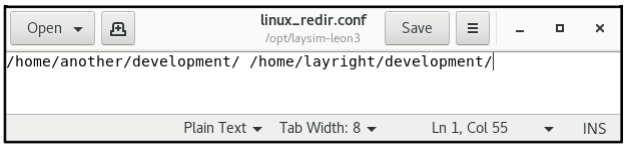
```
-> i
  NAME         ENTRY       TID    PRI   STATUS      PC       SP     ERRNO  CPU #
---------- ------------ -------- --- ---------- -------- -------- ------- -----
tJobTask    166ca0       200efee0   0 PEND         1bc88c 200f3e38       0     -
tExcTask    166480         2934c8   0 PEND         1bc88c   297648       0     -
tLogTask    logTask      200f04e0   0 PEND         1ba768 2010bb60       0     -
tShell0     shellTask    201aa6e8   1 READY        1ca110 201b0af8       0     1
tErfTask    e7f64        20005c50  10 PEND         1bde00 20008e30       0     -
tVxdbgTask  123bd4       2014e838  25 PEND         1bc88c 20151de8       0     -
tNet0       ipcomNetTask 2000dff0  50 PEND         1bd058 20014e50       0     -
tAioIoTask> aioIoTask    20118970  50 PEND         1bde00 2012de30       0     -
tAioIoTask> aioIoTask    20125020  50 PEND         1bde00 2013ae30       0     -
tNetConf    20d1c        2018c530  50 PEND         1bc88c 2018fb40       0     -
tAioWait    aioWaitTask  20118478  51 PEND         1bc88c 20120db0       0     -
miiBusMoni> 1ac148       20145a98 252 DELAY        1c7d60 20148eb8       0     -
tIdleTask0  idleTaskEnt>   2a2000 287 READY        142be4   2a1ed0       0     0
tIdleTask1  idleTaskEnt>   2ae000 287 READY        142be4   2aded0       0     -
value = 0 = 0x0
->
```

■ `quit` : Quit `laysim` CLI version.

3.4 Directory Redirect for Debugging

If an executable which was built from the another platform or has separated source codes, the directory redirection of `laysim-gr712rc` allows to debug it at the source code level. For example, `RTEMS5` examples which were built by another users and their original base directory is '`/home/another/development/`'. After all source codes are copied to '`/home/layright/development`' in Linux, you can set the information of directory redirection in `linux_reconf.conf` as shown in Figure 3-46. It supports up to 10 directory redirection.

Figure 3-46. Set Directory Redirection



4. Loadable Modules

laysim-GR712RC supports loadable modules for SPI, I2C, GPIO, UART, I/O, SpW Nodes, 1553B RTs and CAN Nodes with CAN bus simulation. The evaluation version of laysim-GR712RC provides only the pre-built loadable modules. It is possible to build loadable modules with source codes, and the detailed information can be provided according to the separate license agreement (Professional version of laysim-GR712RC). Detailed information can be found in [laysim-GR712RC-005] laysim-GR712RC User Loadable Modules - User's Manual.

4.1 Loadable I/O Module

The loadable I/O module has 4 components

1) I/O DMA ENGINE (0x20000000 ~ 0x2000000F)
   - Simple example for demonstrating DMA read/write operation
   - Raise GPIO#0 port 3 interrupt after completion
2) I/O C1553BRM MODEL (0x21000000 ~ 0x2100007F)
   - Actel Core 1553BRM
   - Raise GPIO#0 port 3 interrupt
   - 1553B RT traffic monitoring & RT simulation
3) I/O Shared Memory (0x22000000 ~ 0x2201FFFF, 128KB)
   - Shared memory for C1553BRM
4) Board Reset Register (0x20000100)

4.1.1 Actel Core 1553B on I/O Area

Figure 4-1 shows the result of user-io-c1553brm with loadable I/O module. The loadable I/O module can be loaded with '-mod ./Loadable/loadable-IO-Modules/laysim_user_io.so' on start-up. This example is to test the simple BRM interface on B1553BRM with RT simulation. The 128KB SRAM in laysim I/O module is used for 1553B shared memory, and it also supports 1553B traffic monitor.

Figure 4-1. user-io-c1553brm with loadable I/O module

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS user-io]$ laysim-gr712rc-dbt-cli -r -core0 user-io-c1553brm.elf -mod ../Loadable/loadable-IO-Modules/laysim_user_io.so
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ../Loadable/loadable-IO-Modules/laysim_user_io.so

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - user-io-c1553brm.elf
   - .text, addr: 0x40000000, size 65104 bytes
   - .rodata, addr: 0x4000FE50, size 4720 bytes
   - .data, addr: 0x400110C0, size 1560 bytes
   - .bss, addr: 0x400116D8, size 432 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000


Command Block #0
1) Control Word : 0x46FE
  - OPCODE : 0x4 -> Execute Block - Continue (1553B Op.)
  - 1 Retry
  - Channel A
  - No RT-to-RT
  - Condition 7 (No Response) is Enabled
  - Condition 6 (Message Error) is Enabled
  - Condition 5 (Busy bit) is Enabled
  - Condition 4 (Terminal Flag) is Enabled
  - Condition 3 (Subsystem Fail) is Enabled
  - Condition 2 (Instrumentation) is Enabled
  - Condition 1 (Service Request) is Enabled
  - BAME : 0
2) Command Word 1 : 0x0820
  - (01-R-01-32)
3) Command Word 2 : 0x0000
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\user-io> laysim-gr712rc-dbt-cli -r -core0 user-io-c1553brm.elf -mod ..\Loadable\loadable-IO-Modules\laysim_user_io-dbt-cli.dll
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ..\Loadable\loadable-IO-Modules\laysim_user_io-dbt-cli.dll

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - user-io-c1553brm.elf
   - .text, addr: 0x40000000, size 65104 bytes
   - .rodata, addr: 0x4000FE50, size 4720 bytes
   - .data, addr: 0x400110C0, size 1560 bytes
   - .bss, addr: 0x400116D8, size 432 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000


Command Block #0
1) Control Word : 0x46FE
  - OPCODE : 0x4 -> Execute Block - Continue (1553B Op.)
  - 1 Retry
  - Channel A
  - No RT-to-RT
  - Condition 7 (No Response) is Enabled
  - Condition 6 (Message Error) is Enabled
  - Condition 5 (Busy bit) is Enabled
  - Condition 4 (Terminal Flag) is Enabled
  - Condition 3 (Subsystem Fail) is Enabled
  - Condition 2 (Instrumentation) is Enabled
  - Condition 1 (Service Request) is Enabled
  - BAME : 0
2) Command Word 1 : 0x0820
  - (01-R-01-32)
3) Command Word 2 : 0x0000
```

```
4) Data Pointer : 0x0640 (0x22000C80)
   - BC -> RT will be sent :
    0x0000 0x0001 0x0002 0x0003 0x0004 0x0005 0x0006 0x0007
    0x0008 0x0009 0x000A 0x000B 0x000C 0x000D 0x000E 0x000F
    0x0010 0x0011 0x0012 0x0013 0x0014 0x0015 0x0016 0x0017
    0x0018 0x0019 0x001A 0x001B 0x001C 0x001D 0x001E 0x001F
5) Status Word 1 : 0x0000
6) Status Word 2 : 0x0000
7) Branch Address : 0x0000 (0x22000000)
8) Timer Value : 0x0000

Command Block #1
1) Control Word : 0x46FE
  - OPCODE : 0x4 -> Execute Block - Continue (1553B Op.)
  - 1 Retry
  - Channel A
  - No RT-to-RT
  - Condition 7 (No Response) is Enabled
  - Condition 6 (Message Error) is Enabled
  - Condition 5 (Busy bit) is Enabled
  - Condition 4 (Terminal Flag) is Enabled
  - Condition 3 (Subsystem Fail) is Enabled
  - Condition 2 (Instrumentation) is Enabled
  - Condition 1 (Service Request) is Enabled
  - BAME : 0
2) Command Word 1 : 0x0C40
  - (01-T-02-32)
3) Command Word 2 : 0x0000
4) Data Pointer : 0x0660 (0x22000CC0)
5) Status Word 1 : 0x0000
6) Status Word 2 : 0x0000
7) Branch Address : 0x0000 (0x22000000)
8) Timer Value : 0x0000

Command Block #2
1) Control Word : 0x0000
  - OPCODE : 0x0 -> EOL
  - 4 Retry
  - Channel B
  - No RT-to-RT
  - Condition 7 (No Response) is Disabled
  - Condition 6 (Message Error) is Disabled
  - Condition 5 (Busy bit) is Disabled
  - Condition 4 (Terminal Flag) is Disabled
  - Condition 3 (Subsystem Fail) is Disabled
  - Condition 2 (Instrumentation) is Disabled
  - Condition 1 (Service Request) is Disabled
  - BAME : 0
2) Command Word 1 : 0x0000
3) Command Word 2 : 0x0000
4) Data Pointer : 0x0680 (0x22000D00)
5) Status Word 1 : 0x0000
6) Status Word 2 : 0x0000
7) Branch Address : 0x0000 (0x220
(BUS A) : BC -> RT            Msg No : 0 [66457075]
Cmnd:   0820    (01-R-01-32)     RT1-SA01
Data:   0000  0001  0002  0003  0004  0005  0006  0007
        0008  0009  000A  000B  000C  000D  000E  000F
        0010  0011  0012  0013  0014  0015  0016  0017
        0018  0019  001A  001B  001C  001D  001E  001F00
Status: 0800
------------------------------------------------------------

(BUS A) : RT -> BC            Msg No : 1 [66525823]
Cmnd:   0C40    (01-T-02-32)     RT1-SA0200
Status: 0800
Data:   0040  0041  0042  0043  0044  0045  0046  0047
        0048  0049  004A  004B  004C  004D  004E  004F
        0050  0051  0052  0053  0054  0055  0056  0057
        0058  0059  005A  005B  005C  005D  005E  005F
------------------------------------------------------------
0)
8) Timer Value : 0x0000


Register #4 (Pending Interrupt, 0x00000020) : 0x00000000
  - bit 15 - DMAF (DMA Fail Interrupt)   : 0 (Not Occurred)
  - bit 14 - WRAPF (Wrap Fail Interrupt) : 0 (Not Occurred)
  - bit 13 - TAPF (RT Only)              : 0 (Not Occurred)
  - bit 12 - Reserved                    : 0
  - bit 11 - MERR (Message Error Int.)   : 0 (Not Occurred)
  - bit 10 - SUBAD (RT Only)             : 0 (Not Occurred)
  - bit 09 - BDRCV (RT Only)             : 0 (Not Occurred)
  - bit 08 - IXEQ0 (RT Only)             : 0 (Not Occurred)
  - bit 07 - ILLCMD (RT Only)            : 0 (Not Occurred)
  - bit 06 - Reserved                    : 0
  - bit 05 - EOL (End of List Interrupt) : 1 (Occurred)
  - bit 04 - ILLCMD (Illogical Com. Int) : 0 (Not Occurred)
  - bit 03 - ILLOP (Illogical Opc. Int.) : 0 (Not Occurred)
  - bit 02 - RTF (Retry Fail. Interrupt) : 0 (Not Occurred)
  - bit 01 - CBA (Command Blk Acc. Int.) : 0 (Not Occurred)
  - bit 00 - MBC (BM Only)               : 0 (Not Occurred)

Result Report - Command Block #0
1) Control Word : 0x46FE
  - OPCODE : 0x4 -> Execute Block - Continue (1553B Op.)
  - 1 Retry
  - Channel A
  - No RT-to-RT
  - Condition 7 (No Response) is Enabled
  - Condition 6 (Message Error) is Enabled
  - Condition 5 (Busy Bit) is Enabled
  - Condition 4 (Terminal Flag) is Enabled
  - Condition 3 (Subsystem Fail) is Enabled
  - Condition 2 (Instrumentation) is Enabled
  - Condition 1 (Service Request) is Enabled
  - BAME : 0 (No Error)
2) Command Word 1 : 0x0820
  - (01-R-01-32)
3) Command Word 2 : 0x0000
4) Data Pointer : 0x0640 (0x22000C80)
   - BC -> RT was sent :
    0x0000 0x0001 0x0002 0x0003 0x0004 0x0005 0x0006 0x0007
    0x0008 0x0009 0x000A 0x000B 0x000C 0x000D 0x000E 0x000F
    0x0010 0x0011 0x0012 0x0013 0x0014 0x0015 0x0016 0x0017
    0x0018 0x0019 0x001A 0x001B 0x001C 0x001D 0x001E 0x001F
5) Received Status Word 1 : 0x0800
    - bit 11-15 (RT ADDRESSS)        : 1
    - bit 10    (MESSAGE ERROR)      : 0 (Not Set)
    - bit 09    (INSTRUMENTATION)    : 0 (Not Set)
```


```
4) Data Pointer : 0x0640 (0x22000C80)
   - BC -> RT will be sent :
    0x0000 0x0001 0x0002 0x0003 0x0004 0x0005 0x0006 0x0007
    0x0008 0x0009 0x000A 0x000B 0x000C 0x000D 0x000E 0x000F
    0x0010 0x0011 0x0012 0x0013 0x0014 0x0015 0x0016 0x0017
    0x0018 0x0019 0x001A 0x001B 0x001C 0x001D 0x001E 0x001F
5) Status Word 1 : 0x0000
6) Status Word 2 : 0x0000
7) Branch Address : 0x0000 (0x22000000)
8) Timer Value : 0x0000

Command Block #1
1) Control Word : 0x46FE
  - OPCODE : 0x4 -> Execute Block - Continue (1553B Op.)
  - 1 Retry
  - Channel A
  - No RT-to-RT
  - Condition 7 (No Response) is Enabled
  - Condition 6 (Message Error) is Enabled
  - Condition 5 (Busy bit) is Enabled
  - Condition 4 (Terminal Flag) is Enabled
  - Condition 3 (Subsystem Fail) is Enabled
  - Condition 2 (Instrumentation) is Enabled
  - Condition 1 (Service Request) is Enabled
  - BAME : 0
2) Command Word 1 : 0x0C40
  - (01-T-02-32)
3) Command Word 2 : 0x0000
4) Data Pointer : 0x0660 (0x22000CC0)
5) Status Word 1 : 0x0000
6) Status Word 2 : 0x0000
7) Branch Address : 0x0000 (0x22000000)
8) Timer Value : 0x0000

Command Block #2
1) Control Word : 0x0000
  - OPCODE : 0x0 -> EOL
  - 4 Retry
  - Channel B
  - No RT-to-RT
  - Condition 7 (No Response) is Disabled
  - Condition 6 (Message Error) is Disabled
  - Condition 5 (Busy bit) is Disabled
  - Condition 4 (Terminal Flag) is Disabled
  - Condition 3 (Subsystem Fail) is Disabled
  - Condition 2 (Instrumentation) is Disabled
  - Condition 1 (Service Request) is Disabled
  - BAME : 0
2) Command Word 1 : 0x0000
3) Command Word 2 : 0x0000
4) Data Pointer : 0x0680 (0x22000D00)
5) Status Word 1 : 0x0000
6) Status Word 2 : 0x0000
7) Branch Address : 0x0000 (0x220
(BUS A) : BC -> RT       Msg No : 0 [66457075]
Cmnd:   0820    (01-R-01-32)     RT1-SA01
Data:   0000  0001  0002  0003  0004  0005  0006  0007
        0008  0009  000A  000B  000C  000D  000E  000F
        0010  0011  0012  0013  0014  0015  0016  0017
        0018  0019  001A  001B  001C  001D  001E  001F00
Status: 0800
------------------------------------------------------------

(BUS A) : RT -> BC       Msg No : 1 [66525823]
Cmnd:   0C40    (01-T-02-32)     RT1-SA0200
Status: 0800
Data:   0040  0041  0042  0043  0044  0045  0046  0047
        0048  0049  004A  004B  004C  004D  004E  004F
        0050  0051  0052  0053  0054  0055  0056  0057
        0058  0059  005A  005B  005C  005D  005E  005F
------------------------------------------------------------
0)
8) Timer Value : 0x0000


Register #4 (Pending Interrupt, 0x00000020) : 0x00000000
  - bit 15 - DMAF (DMA Fail Interrupt)   : 0 (Not Occurred)
  - bit 14 - WRAPF (Wrap Fail Interrupt) : 0 (Not Occurred)
  - bit 13 - TAPF (RT Only)              : 0 (Not Occurred)
  - bit 12 - Reserved                    : 0
  - bit 11 - MERR (Message Error Int.)   : 0 (Not Occurred)
  - bit 10 - SUBAD (RT Only)             : 0 (Not Occurred)
  - bit 09 - BDRCV (RT Only)             : 0 (Not Occurred)
  - bit 08 - IXEQ0 (RT Only)             : 0 (Not Occurred)
  - bit 07 - ILLCMD (RT Only)            : 0 (Not Occurred)
  - bit 06 - Reserved                    : 0
  - bit 05 - EOL (End of List Interrupt) : 1 (Occurred)
  - bit 04 - ILLCMD (Illogical Com. Int) : 0 (Not Occurred)
  - bit 03 - ILLOP (Illogical Opc. Int.) : 0 (Not Occurred)
  - bit 02 - RTF (Retry Fail. Interrupt) : 0 (Not Occurred)
  - bit 01 - CBA (Command Blk Acc. Int.) : 0 (Not Occurred)
  - bit 00 - MBC (BM Only)               : 0 (Not Occurred)

Result Report - Command Block #0
1) Control Word : 0x46FE
  - OPCODE : 0x4 -> Execute Block - Continue (1553B Op.)
  - 1 Retry
  - Channel A
  - No RT-to-RT
  - Condition 7 (No Response) is Enabled
  - Condition 6 (Message Error) is Enabled
  - Condition 5 (Busy Bit) is Enabled
  - Condition 4 (Terminal Flag) is Enabled
  - Condition 3 (Subsystem Fail) is Enabled
  - Condition 2 (Instrumentation) is Enabled
  - Condition 1 (Service Request) is Enabled
  - BAME : 0 (No Error)
2) Command Word 1 : 0x0820
  - (01-R-01-32)
3) Command Word 2 : 0x0000
4) Data Pointer : 0x0640 (0x22000C80)
   - BC -> RT was sent :
    0x0000 0x0001 0x0002 0x0003 0x0004 0x0005 0x0006 0x0007
    0x0008 0x0009 0x000A 0x000B 0x000C 0x000D 0x000E 0x000F
    0x0010 0x0011 0x0012 0x0013 0x0014 0x0015 0x0016 0x0017
    0x0018 0x0019 0x001A 0x001B 0x001C 0x001D 0x001E 0x001F
5) Received Status Word 1 : 0x0800
    - bit 11-15 (RT ADDRESSS)        : 1
    - bit 10    (MESSAGE ERROR)      : 0 (Not Set)
    - bit 09    (INSTRUMENTATION)    : 0 (Not Set)
```

```
    - bit 08    (SERVICE REQUEST)    : 0 (Not Set)
    - bit 05-07 (RESERVED)           : 0
    - bit 04    (BROADCAST CMD RCVED): 0 (Not Set)
    - bit 03    (BUSY)               : 0 (Not Set)
    - bit 02    (SUBSYS FLAG)        : 0 (Not Set)
    - bit 01    (DYNAMIC BUS CTRL)   : 0 (Not Set)
    - bit 00    (TERMINAL FLAG)      : 0 (Not Set)
6) Received Status Word 2 : 0x0000
7) Branch Address : 0x0000 (0x22000000)
8) Timer Value : 0x0000

Result Report - Command Block #1
1) Control Word : 0x46FE
  - OPCODE : 0x4 -> Execute Block - Continue (1553B Op.)
  - 1 Retry
  - Channel A
  - No RT-to-RT
  - Condition 7 (No Response) is Enabled
  - Condition 6 (Message Error) is Enabled
  - Condition 5 (Busy Bit) is Enabled
  - Condition 4 (Terminal Flag) is Enabled
  - Condition 3 (Subsystem Fail) is Enabled
  - Condition 2 (Instrumentation) is Enabled
  - Condition 1 (Service Request) is Enabled
  - BAME : 0 (No Error)
2) Command Word 1 : 0x0C40
  - (01-T-02-32)
3) Command Word 2 : 0x0000
4) Data Pointer : 0x0660 (0x22000CC0)
   - BC <- RT was received :
    0x0040 0x0041 0x0042 0x0043 0x0044 0x0045 0x0046 0x0047
    0x0048 0x0049 0x004A 0x004B 0x004C 0x004D 0x004E 0x004F
    0x0050 0x0051 0x0052 0x0053 0x0054 0x0055 0x0056 0x0057
    0x0058 0x0059 0x005A 0x005B 0x005C 0x005D 0x005E 0x005F
5) Received Status Word 1 : 0x0800
    - bit 11-15 (RT ADDRESSS)        : 1
    - bit 10    (MESSAGE ERROR)      : 0 (Not Set)
    - bit 09    (INSTRUMENTATION)    : 0 (Not Set)
    - bit 08    (SERVICE REQUEST)    : 0 (Not Set)
    - bit 05-07 (RESERVED)           : 0
    - bit 04    (BROADCAST CMD RCVED): 0 (Not Set)
    - bit 03    (BUSY)               : 0 (Not Set)
    - bit 02    (SUBSYS FLAG)        : 0 (Not Set)
    - bit 01    (DYNAMIC BUS CTRL)   : 0 (Not Set)
    - bit 00    (TERMINAL FLAG)      : 0 (Not Set)
6) Received Status Word 2 : 0x0000
7) Branch Address : 0x0000 (0x22000000)
8) Timer Value : 0x0000

Result Report - Command Block #2
1) Control Word : 0x0000
  - OPCODE : 0x0 -> EOL
  - 4 Retry
  - Channel B
  - No RT-to-RT
  - Condition 7 (No Response) is Disabled
  - Condition 6 (Message Error) is Disabled
  - Condition 5 (Busy Bit) is Disabled
  - Condition 4 (Terminal Flag) is Disabled
  - Condition 3 (Subsystem Fail) is Disabled
  - Condition 2 (Instrumentation) is Disabled
  - Condition 1 (Service Request) is Disabled
  - BAME : 0 (No Error)
2) Command Word 1 : 0x0000
3) Command Word 2 : 0x0000
4) Data Pointer : 0x0680 (0x22000D00)
5) Received Status Word 1 : 0x0000
6) Received Status Word 2 : 0x0000
7) Branch Address : 0x0000 (0x22000000)
8) Timer Value : 0x0000
Test done!
CORE#0 halts because of Error Mode [0.23371sec/204518680, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 204518680
   Instructions : 65449957
   CORE#0 performance (100MHz) : 32.00 MIPS
   Simulated time              : 2.05 s

 Performance of laysim-gr712rc
   Real-time performance       : 875.10 %
   laysim-gr712rc performance  : 280.05 MIPS
   Wall clock                  : 0.23 s
[layright@CentOS user-io]$
```

```
    - bit 08    (SERVICE REQUEST)    : 0 (Not Set)
    - bit 05-07 (RESERVED)           : 0
    - bit 04    (BROADCAST CMD RCVED): 0 (Not Set)
    - bit 03    (BUSY)               : 0 (Not Set)
    - bit 02    (SUBSYS FLAG)        : 0 (Not Set)
    - bit 01    (DYNAMIC BUS CTRL)   : 0 (Not Set)
    - bit 00    (TERMINAL FLAG)      : 0 (Not Set)
6) Received Status Word 2 : 0x0000
7) Branch Address : 0x0000 (0x22000000)
8) Timer Value : 0x0000

Result Report - Command Block #1
1) Control Word : 0x46FE
  - OPCODE : 0x4 -> Execute Block - Continue (1553B Op.)
  - 1 Retry
  - Channel A
  - No RT-to-RT
  - Condition 7 (No Response) is Enabled
  - Condition 6 (Message Error) is Enabled
  - Condition 5 (Busy Bit) is Enabled
  - Condition 4 (Terminal Flag) is Enabled
  - Condition 3 (Subsystem Fail) is Enabled
  - Condition 2 (Instrumentation) is Enabled
  - Condition 1 (Service Request) is Enabled
  - BAME : 0 (No Error)
2) Command Word 1 : 0x0C40
  - (01-T-02-32)
3) Command Word 2 : 0x0000
4) Data Pointer : 0x0660 (0x22000CC0)
   - BC <- RT was received :
    0x0040 0x0041 0x0042 0x0043 0x0044 0x0045 0x0046 0x0047
    0x0048 0x0049 0x004A 0x004B 0x004C 0x004D 0x004E 0x004F
    0x0050 0x0051 0x0052 0x0053 0x0054 0x0055 0x0056 0x0057
    0x0058 0x0059 0x005A 0x005B 0x005C 0x005D 0x005E 0x005F
5) Received Status Word 1 : 0x0800
    - bit 11-15 (RT ADDRESSS)        : 1
    - bit 10    (MESSAGE ERROR)      : 0 (Not Set)
    - bit 09    (INSTRUMENTATION)    : 0 (Not Set)
    - bit 08    (SERVICE REQUEST)    : 0 (Not Set)
    - bit 05-07 (RESERVED)           : 0
    - bit 04    (BROADCAST CMD RCVED): 0 (Not Set)
    - bit 03    (BUSY)               : 0 (Not Set)
    - bit 02    (SUBSYS FLAG)        : 0 (Not Set)
    - bit 01    (DYNAMIC BUS CTRL)   : 0 (Not Set)
    - bit 00    (TERMINAL FLAG)      : 0 (Not Set)
6) Received Status Word 2 : 0x0000
7) Branch Address : 0x0000 (0x22000000)
8) Timer Value : 0x0000

Result Report - Command Block #2
1) Control Word : 0x0000
  - OPCODE : 0x0 -> EOL
  - 4 Retry
  - Channel B
  - No RT-to-RT
  - Condition 7 (No Response) is Disabled
  - Condition 6 (Message Error) is Disabled
  - Condition 5 (Busy Bit) is Disabled
  - Condition 4 (Terminal Flag) is Disabled
  - Condition 3 (Subsystem Fail) is Disabled
  - Condition 2 (Instrumentation) is Disabled
  - Condition 1 (Service Request) is Disabled
  - BAME : 0 (No Error)
2) Command Word 1 : 0x0000
3) Command Word 2 : 0x0000
4) Data Pointer : 0x0680 (0x22000D00)
5) Received Status Word 1 : 0x0000
6) Received Status Word 2 : 0x0000
7) Branch Address : 0x0000 (0x22000000)
8) Timer Value : 0x0000
Test done!
CORE#0 halts because of Error Mode [0.970644sec/204518680, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 204518680
   Instructions : 65449957
   CORE#0 performance (100MHz) : 32.00 MIPS
   Simulated time              : 2.05 s

 Performance of laysim-gr712rc
   Real-time performance       : 210.70 %
   laysim-gr712rc performance  : 67.43 MIPS
   Wall clock                  : 0.97 s
PS C:\laysim-gr712rc\user-io>
```

4.1.2 Board Reset on I/O Area

Figure 4-2 shows the result of `smp08-reset.rom` with the loadable I/O module. The loadable I/O module can be loaded with '`-mod ./Loadable/loadable-IO/laysim_user_io.so`' on start-up. This example is to test the board reset from the board reset register in I/O area.

Figure 4-2. smp08-reset with loadable I/O module

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS user-io]$ laysim-gr712rc-dbt-cli -r -rom smp08-reset.rom -mod ../Loadable/loadable-IO-Modules/laysim_user_io.so
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ../Loadable/loadable-IO-Modules/laysim_user_io.so

Multicore control : Time quantum (200)
ROM file is loaded - smp08-reset.rom
   - binary file size : 130272 bytes
GR712RC ROM file is loaded, starts from ROM
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\user-io> laysim-gr712rc-dbt-cli -r -rom smp08-reset.rom -mod ..\Loadable\loadable-IO-Modules\laysim_user_io-dbt-cli.dll
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ..\Loadable\loadable-IO-Modules\laysim_user_io-dbt-cli.dll

Multicore control : Time quantum (200)
ROM file is loaded - smp08-reset.rom
   - binary file size : 130272 bytes
GR712RC ROM file is loaded, starts from ROM
```

```
MKPROM2 boot loader v2.0.65
Copyright Cobham Gaisler AB - all rights reserved

system clock   : 100.0 MHz
baud rate      : 38343 baud
prom           : 8192 K, (15/15) ws (r/w)
sram           : 8192 K, 1 bank(s), 2/2 ws (r/w)

decompressing .text to 0x40000000
decompressing .rtemsroset to 0x400333a0
decompressing .data to 0x4003b480

starting smp08-reset.elf


*** BEGIN OF TEST SMP 8 ***
*** TEST VERSION: 5.0.0.cba7f0485d9c8a37a889586b188070f365106065-modified
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 10.2.0
  CPU 1 running task Init - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA03 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA04 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA05 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA06 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:05   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA03 - rtems_clock_get_tod 09:00:15   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:15   12/31/1988


MKPROM2 boot loader v2.0.65
Copyright Cobham Gaisler AB - all rights reserved

system clock   : 100.0 MHz
baud rate      : 38343 baud
prom           : 8192 K, (15/15) ws (r/w)
sram           : 8192 K, 1 bank(s), 2/2 ws (r/w)

decompressing .text to 0x40000000
decompressing .rtemsroset to 0x400333a0
decompressing .data to 0x4003b480

starting smp08-reset.elf


*** BEGIN OF TEST SMP 8 ***
*** TEST VERSION: 5.0.0.cba7f0485d9c8a37a889586b188070f365106065-modified
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 10.2.0
  CPU 1 running task Init - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA03 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA04 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA05 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA06 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:05   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA03 - rtems_clock_get_tod 09:00:15   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:15   12/31/1988


MKPROM2 boot loader v2.0.65
Copyright Cobham Gaisler AB - all rights reserved

system clock   : 100.0 MHz
baud rate      : 38343 baud
prom           : 8192 K, (15/15) ws (r/w)
sram           : 8192 K, 1 bank(s), 2/2 ws (r/w)

decompressing .text to 0x40000000
decompressing .rtemsroset to 0x400333a0
decompressing .data to 0x4003b480

starting smp08-reset.elf


*** BEGIN OF TEST SMP 8 ***
*** TEST VERSION: 5.0.0.cba7f0485d9c8a37a889586b188070f365106065-modified
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 10.2.0
  CPU 1 running task Init - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA03 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA04 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA05 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA06 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:05   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA03 - rtems_clock_get_tod 09:00:15   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:15   12/31/1988
...
...
```

```
MKPROM2 boot loader v2.0.65
Copyright Cobham Gaisler AB - all rights reserved

system clock   : 100.0 MHz
baud rate      : 38343 baud
prom           : 8192 K, (15/15) ws (r/w)
sram           : 8192 K, 1 bank(s), 2/2 ws (r/w)

decompressing .text to 0x40000000
decompressing .rtemsroset to 0x400333a0
decompressing .data to 0x4003b480

starting smp08-reset.elf


*** BEGIN OF TEST SMP 8 ***
*** TEST VERSION: 5.0.0.cba7f0485d9c8a37a889586b188070f365106065-modified
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 10.2.0
  CPU 1 running task Init - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA03 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA04 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA05 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA06 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:05   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA03 - rtems_clock_get_tod 09:00:15   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:15   12/31/1988


MKPROM2 boot loader v2.0.65
Copyright Cobham Gaisler AB - all rights reserved

system clock   : 100.0 MHz
baud rate      : 38343 baud
prom           : 8192 K, (15/15) ws (r/w)
sram           : 8192 K, 1 bank(s), 2/2 ws (r/w)

decompressing .text to 0x40000000
decompressing .rtemsroset to 0x400333a0
decompressing .data to 0x4003b480

starting smp08-reset.elf


*** BEGIN OF TEST SMP 8 ***
*** TEST VERSION: 5.0.0.cba7f0485d9c8a37a889586b188070f365106065-modified
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 10.2.0
  CPU 1 running task Init - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA03 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA04 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA05 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA06 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:05   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA03 - rtems_clock_get_tod 09:00:15   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:15   12/31/1988


MKPROM2 boot loader v2.0.65
Copyright Cobham Gaisler AB - all rights reserved

system clock   : 100.0 MHz
baud rate      : 38343 baud
prom           : 8192 K, (15/15) ws (r/w)
sram           : 8192 K, 1 bank(s), 2/2 ws (r/w)

decompressing .text to 0x40000000
decompressing .rtemsroset to 0x400333a0
decompressing .data to 0x4003b480

starting smp08-reset.elf


*** BEGIN OF TEST SMP 8 ***
*** TEST VERSION: 5.0.0.cba7f0485d9c8a37a889586b188070f365106065-modified
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 10.2.0
  CPU 1 running task Init - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA03 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA04 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA05 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 0 running task TA06 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:00   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:05   12/31/1988
  CPU 1 running task TA01 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA02 - rtems_clock_get_tod 09:00:10   12/31/1988
  CPU 1 running task TA03 - rtems_clock_get_tod 09:00:15   12/31/1988
  CPU 0 running task TA01 - rtems_clock_get_tod 09:00:15   12/31/1988
...
...
```

4.1.3 DMA Operation on I/O Area

Figure 4-2 shows the result of `user-io-dma` with the loadable I/O module. The loadable I/O module can be loaded with '`-mod ./Loadable/loadable-IO-Modules/laysim_user_io.so`' on start-up. This example is to test the simple DMA read/write operation in I/O area. It copies the SRAM data to I/O shared memory using DMA and it raises the GPIO#0 port 3 interrupt.

Figure 4-3. user-io-dma with loadable I/O module

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS user-io]$ laysim-gr712rc-dbt-cli -r -core0 user-io-dma.elf -mod ../Loa
dable/loadable-IO-Modules/laysim_user_io.so
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ../Loadable/loadable-IO-Modules/laysim_user_io.so

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - user-io-dma.elf
   - .text, addr: 0x40000000, size 58640 bytes
   - .rodata, addr: 0x4000E510, size 1664 bytes
   - .data, addr: 0x4000EB90, size 1560 bytes
   - .bss, addr: 0x4000F1A8, size 464 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

22000000 : DEAD0000
22000004 : DEAD0001
22000008 : DEAD0002
2200000C : DEAD0003
22000010 : DEAD0004
22000014 : DEAD0005
22000018 : DEAD0006
2200001C : DEAD0007
22000020 : DEAD0008
22000024 : DEAD0009
Test done!
CORE#0 halts because of Error Mode [0.009054sec/6327347, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 6327347
   Instructions : 2061347
   CORE#0 performance (100MHz) : 32.58 MIPS
   Simulated time              : 63.27 ms

 Performance of laysim-gr712rc
   Real-time performance       : 698.85 %
   laysim-gr712rc performance  : 227.67 MIPS
   Wall clock                  : 0.01 s
[layright@CentOS user-io]$
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\user-io> laysim-gr712rc-dbt-cli -r -core0 user-io-dma.elf -mod ..
\Loadable\loadable-IO-Modules\laysim_user_io-dbt-cli.dll
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ..\Loadable\loadable-IO-Modules\laysim_user_io-dbt-cli.dll

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - user-io-dma.elf
   - .text, addr: 0x40000000, size 58640 bytes
   - .rodata, addr: 0x4000E510, size 1664 bytes
   - .data, addr: 0x4000EB90, size 1560 bytes
   - .bss, addr: 0x4000F1A8, size 464 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

22000000 : DEAD0000
22000004 : DEAD0001
22000008 : DEAD0002
2200000C : DEAD0003
22000010 : DEAD0004
22000014 : DEAD0005
22000018 : DEAD0006
2200001C : DEAD0007
22000020 : DEAD0008
22000024 : DEAD0009
Test done!
CORE#0 halts because of Error Mode [0.0281099sec/6327347, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 6327347
   Instructions : 2061347
   CORE#0 performance (100MHz) : 32.58 MIPS
   Simulated time              : 63.27 ms

 Performance of laysim-gr712rc
   Real-time performance       : 225.09 %
   laysim-gr712rc performance  : 73.33 MIPS
   Wall clock                  : 0.03 s
PS C:\laysim-gr712rc\user-io>
```

4.2 Loadable SpW Module

`laysim-GR712RC` supports the loadable SpW module by using '`-mod MODULE`' on start-up. Figure 4-4 shows the result of `rtems-rmap` with the loadable SpW module. The loadable SpW module can be loaded with '`-mod ./Loadable/loadable-SpW-Nodes/laysim_user_spw_node.so`' on start-up.

Figure 4-4. SpW Node Simulation using loadable SpW module

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS RCC-1.3.1]$ laysim-gr712rc-dbt-cli -r -core0 rtems-rmap.elf -mod ../Lo
adable/loadable-SpW-Nodes/laysim_user_spw_node_module1.so
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ../Loadable/loadable-SpW-Nodes/laysim_user_spw_node_module
1.so

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - rtems-rmap.elf
   - .text, addr: 0x40000000, size 237664 bytes
   - .rtemsroset, addr: 0x4003A060, size 176 bytes
   - .rtemsstack, addr: 0x4003A140, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x4003C140, size 8288 bytes
   - .bss, addr: 0x4003E1C0, size 106640 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

#### To change configuration of this example, you can set a breakpoint
#### on label <rmap_conf_label> and modify the following global variables:
## remote_base_address: 0x40000000 /* Base address on Remote. This is the address to wh
ich the data is copied. */
## remote_dst_address: 0xfe /* SpW Destination address. */
## remote_dst_key: 0x00/* SpW Destination Key */
## source_src_address: 0x20 /* SpW Source address */
## rmap_dev_idx: 0 /* This parameter chooses which initiator GRSPW2 core is used */
## route_entry { nr; /* Number of addresses in dstadr array */, dstadr[16];/* Path Addr
esses */}
## routetab[ROUTE_TO]={2,{1,5,0, ...}
## routetab[ROUTE_FROM]={2,{1,9,0, ...}
## path_addressing: 0 /* Use path addressing (!=0) or not (0) */
## rmap_pkt_size: 992
## rmap_data_bytes: 4096 /* Data bytes to send over RMAP */
#### RTEMS RMAP example

Setting up SpaceWire router
Configuring Router
Failed to open SpW Router
Router_configure: Error -1
Failed router initialization.
Limiting to 1 SpaceWire devices
Initializing SpaceWire device 0
 After Link Start: 5

Test built for at least 1 AMBA ports.
Note that the SpW-links will not be used.
There are 6 GRSPW cores in the system.
System clock: 10000 us / tick
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\RCC-1.3.1> laysim-gr712rc-dbt-cli -r -core0 rtems-rmap.elf -mod ..
\Loadable\loadable-SpW-Nodes\laysim_user_spw_node_module1-dbt-cli.dll
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ..\Loadable\loadable-SpW-Nodes\laysim_user_spw_node_module1
-dbt-cli.dll

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - rtems-rmap.elf
   - .text, addr: 0x40000000, size 237664 bytes
   - .rtemsroset, addr: 0x4003A060, size 176 bytes
   - .rtemsstack, addr: 0x4003A140, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x4003C140, size 8288 bytes
   - .bss, addr: 0x4003E1C0, size 106640 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000

#### To change configuration of this example, you can set a breakpoint
#### on label <rmap_conf_label> and modify the following global variables:
## remote_base_address: 0x40000000 /* Base address on Remote. This is the address to wh
ich the data is copied. */
## remote_dst_address: 0xfe /* SpW Destination address. */
## remote_dst_key: 0x00/* SpW Destination Key */
## source_src_address: 0x20 /* SpW Source address */
## rmap_dev_idx: 0 /* This parameter chooses which initiator GRSPW2 core is used */
## route_entry { nr; /* Number of addresses in dstadr array */, dstadr[16];/* Path Addr
esses */}
## routetab[ROUTE_TO]={2,{1,5,0, ...}
## routetab[ROUTE_FROM]={2,{1,9,0, ...}
## path_addressing: 0 /* Use path addressing (!=0) or not (0) */
## rmap_pkt_size: 992
## rmap_data_bytes: 4096 /* Data bytes to send over RMAP */
#### RTEMS RMAP example

Setting up SpaceWire router
Configuring Router
Failed to open SpW Router
Router_configure: Error -1
Failed router initialization.
Limiting to 1 SpaceWire devices
Initializing SpaceWire device 0
 After Link Start: 5

Test built for at least 1 AMBA ports.
Note that the SpW-links will not be used.
There are 6 GRSPW cores in the system.
System clock: 10000 us / tick
```

```
Starting GRSPW0: DMA Started Successfully
RMAP connection is working
Initializing data array


Starting transmission type 1: one packet at a time

RMAP write cycle completed in 0.000588 seconds
RMAP read cycle completed in 0.000590 seconds
Data bytes sent: 4096, Packet size: 992, Packet count: 5
Wrote 4096 bytes to target in 0.000588 seconds. WRITE RATE: 6802.723163 KiB/s
Read 4096 bytes from target in 0.000590 seconds. READ RATE: 6779.700359 KiB/s
Wrote 5 packets to target in 0.000588 seconds. WRITE RATE: 8503.403953 pkt/s
Read 5 packets from target in 0.000590 seconds. READ RATE: 8474.625448 pkt/s


EXAMPLE 1 SUCCESSFULLY COMPLETED.



Starting transmission type 2: all packets at a time

RMAP write cycle completed in 0.000474 seconds
RMAP read cycle completed in 0.000421 seconds
Data bytes sent: 4096, Packet size: 992, Packet count: 5
Wrote 4096 bytes to target in 0.000474 seconds. WRITE RATE: 8439.243461 KiB/s
Read 4096 bytes from target in 0.000421 seconds. READ RATE: 9501.467365 KiB/s
Wrote 5 packets to target in 0.000474 seconds. WRITE RATE: 10549.054326 pkt/s
Read 5 packets from target in 0.000421 seconds. READ RATE: 11876.834206 pkt/s


EXAMPLE 2 SUCCESSFULLY COMPLETED.

CORE#0 halts because of Error Mode [0.049967sec/78601961, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 78601961
   Instructions : 45350689
   CORE#0 performance (100MHz) : 57.70 MIPS
   Simulated time              : 786.02 ms

 Performance of laysim-gr712rc
   Real-time performance       : 1573.08 %
   laysim-gr712rc performance  : 907.61 MIPS
   Wall clock                  : 0.05 s
[layright@CentOS RCC-1.3.1]$
```

```
Starting GRSPW0: DMA Started Successfully
RMAP connection is working
Initializing data array


Starting transmission type 1: one packet at a time

RMAP write cycle completed in 0.000588 seconds
RMAP read cycle completed in 0.000590 seconds
Data bytes sent: 4096, Packet size: 992, Packet count: 5
Wrote 4096 bytes to target in 0.000588 seconds. WRITE RATE: 6802.723163 KiB/s
Read 4096 bytes from target in 0.000590 seconds. READ RATE: 6779.700359 KiB/s
Wrote 5 packets to target in 0.000588 seconds. WRITE RATE: 8503.403953 pkt/s
Read 5 packets from target in 0.000590 seconds. READ RATE: 8474.625448 pkt/s


EXAMPLE 1 SUCCESSFULLY COMPLETED.



Starting transmission type 2: all packets at a time

RMAP write cycle completed in 0.000474 seconds
RMAP read cycle completed in 0.000421 seconds
Data bytes sent: 4096, Packet size: 992, Packet count: 5
Wrote 4096 bytes to target in 0.000474 seconds. WRITE RATE: 8439.243461 KiB/s
Read 4096 bytes from target in 0.000421 seconds. READ RATE: 9501.467365 KiB/s
Wrote 5 packets to target in 0.000474 seconds. WRITE RATE: 10549.054326 pkt/s
Read 5 packets from target in 0.000421 seconds. READ RATE: 11876.834206 pkt/s


EXAMPLE 2 SUCCESSFULLY COMPLETED.

CORE#0 halts because of Error Mode [0.264649sec/78601961, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 78601961
   Instructions : 45350689
   CORE#0 performance (100MHz) : 57.70 MIPS
   Simulated time              : 786.02 ms

 Performance of laysim-gr712rc
   Real-time performance       : 297.00 %
   laysim-gr712rc performance  : 171.36 MIPS
   Wall clock                  : 0.26 s
PS C:\laysim-gr712rc\RCC-1.3.1>
```

4.3 Loadable 1553B RT Module

`laysim-GR712RC` supports the loadable 1553B RT module by using '`-mod MODULE`' on start-up. Figure 4-5 shows the result of `rtems-brm-bc` with the loadable 1553B RT module. The loadable 1553B RT module can be loaded with '`-mod ./Loadable/loadable-1553RT-Modules/laysim_user_1553rt_module.so`' on start-up. This example is to test the Actel Core 1553BRM with RT simulation.

Figure 4-5. 1553B RT Simulation using loadable 1553B RT module

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS RCC-1.3.1]$ laysim-gr712rc-dbt-cli -r -core0 rtems-brm-bc.elf -mod ../
Loadable/loadable-1553RT-Modules/laysim_user_1553rt_module.so
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ../Loadable/loadable-1553RT-Modules/laysim_user_1553rt_modu
le.so

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - rtems-brm-bc.elf
   - .text, addr: 0x40000000, size 229744 bytes
   - .rtemsroset, addr: 0x40038170, size 160 bytes
   - .rtemsstack, addr: 0x40038240, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x4003A240, size 6672 bytes
   - .bss, addr: 0x4003BC80, size 22432 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
******** Starting Gaisler BRM test ********
Starting task 1: BC mode
brmlib_open: Opening driver /dev/b1553brm0
brmlib_open: allocating memory 20
Task1: Setting BC mode
Task1: Setting TX/RX blocking mode
Setting up command list.
-------------  BC: START LIST EXECUTION -------------
Start CMD list processing.
(BUS A) : BC -> RT            Msg No : 0 [9852056]
Cmnd:   0828    (01-R-01-08)     RT1-SA01
Data:   0000  0047  0052  0007  0000  0000  0000  0000
Status: 0800
------------------------------------------------------

(BUS A) : BC -> RT            Msg No : 1 [9872800]
Cmnd:   0848    (01-R-02-08)     RT1-SA02
Data:   0000  0047  0052  0008  0000  0000  0000  0000
Status: 0800
------------------------------------------------------


(BUS A) : BC -> RT            Msg No : 2 [9893546]
Cmnd:   0868    (01-R-03-08)     RT1-SA03
Data:   0000  0047  0052  0009  0000  0000  0000  0000
Status: 0800
------------------------------------------------------
W
(BUS A) : BC -> RT            Msg No : 3 [9914311]
Cmnd:   0889    (01-R-04-09)     RT1-SA04
Data:   0000  0047  0052  000A  0000  0000  0000  0000
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\RCC-1.3.1> laysim-gr712rc-dbt-cli -r -core0 rtems-brm-bc.elf -mod
..\Loadable\loadable-1553RT-Modules\laysim_user_1553rt_module-dbt-cli.dll
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ..\Loadable\loadable-1553RT-Modules\laysim_user_1553rt_modu
le-dbt-cli.dll

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - rtems-brm-bc.elf
   - .text, addr: 0x40000000, size 229744 bytes
   - .rtemsroset, addr: 0x40038170, size 160 bytes
   - .rtemsstack, addr: 0x40038240, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x4003A240, size 6672 bytes
   - .bss, addr: 0x4003BC80, size 22432 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
******** Starting Gaisler BRM test ********
Starting task 1: BC mode
brmlib_open: Opening driver /dev/b1553brm0
brmlib_open: allocating memory 20
Task1: Setting BC mode
Task1: Setting TX/RX blocking mode
Setting up command list.
-------------  BC: START LIST EXECUTION -------------
Start CMD list processing.
(BUS A) : BC -> RT       Msg No : 0 [9852056]
Cmnd:   0828    (01-R-01-08)     RT1-SA01
Data:   0000  0047  0052  0007  0000  0000  0000  0000
Status: 0800
------------------------------------------------------

(BUS A) : BC -> RT       Msg No : 1 [9872800]
Cmnd:   0848    (01-R-02-08)     RT1-SA02
Data:   0000  0047  0052  0008  0000  0000  0000  0000
Status: 0800
------------------------------------------------------


(BUS A) : BC -> RT       Msg No : 2 [9893546]
Cmnd:   0868    (01-R-03-08)     RT1-SA03
Data:   0000  0047  0052  0009  0000  0000  0000  0000
Status: 0800
------------------------------------------------------
W
(BUS A) : BC -> RT       Msg No : 3 [9914311]
Cmnd:   0889    (01-R-04-09)     RT1-SA04
Data:   0000  0047  0052  000A  0000  0000  0000  0000
```

```
        0000
Status: 0800
------------------------------------------------------------
aiting until list processed
List processed.
Sleeping 20s
-------------  BC: START LIST EXECUTION -------------
Start RESULT list processing.
(BUS A) : RT -> BC            Msg No : 4 [2014904315]
Cmnd:   0C28    (01-T-01-08)     RT1-SA01
Status: 0800
Data:   0020  0021  0022  0023  0024  0025  0026  0027
------------------------------------------------------------

(BUS A) : RT -> BC            Msg No : 5 [2014925059]
Cmnd:   0C48    (01-T-02-08)     RT1-SA02
Status: 0800
Data:   0040  0041  0042  0043  0044  0045  0046  0047
------------------------------------------------------------


(BUS A) : RT -> BC            Msg No : 6 [2014945805]
Cmnd:   0C68    (01-T-03-08)     RT1-SA03
Status: 0800
Data:   0060  0061  0062  0063  0064  0065  0066  0067
------------------------------------------------------------
W
(BUS A) : RT -> BC            Msg No : 7 [2014966570]
Cmnd:   0C89    (01-T-04-09)     RT1-SA04
Status: 0800
Data:   0080  0081  0082  0083  0084  0085  0086  0087
        0088
------------------------------------------------------------
aiting until list processed
List processed.
Response to message 1: (len: 8, tsw1: 800, tsw2: 0)
  0x20 (.)0x21 (.)0x22 (.)0x23 (.)0x24 (.)0x25 (.)0x26 (.)0x27 (.)
Response to message 2: (len: 8, tsw1: 800, tsw2: 0)
  0x40 (.)0x41 (A)0x42 (B)0x43 (C)0x44 (D)0x45 (E)0x46 (F)0x47 (G)
Response to message 3: (len: 8, tsw1: 800, tsw2: 0)
  0x60 (.)0x61 (a)0x62 (b)0x63 (c)0x64 (d)0x65 (e)0x66 (f)0x67 (g)
Response to message 4: (len: 9, tsw1: 800, tsw2: 0)
  0x80 (.)0x81 (.)0x82 (.)0x83 (.)0x84 (.)0x85 (.)0x86 (.)0x87 (.)0x88 (.)
------------------------------------------------------
Sleeping 15s
-------------  BC: START LIST EXECUTION -------------
Start CMD list processing.
(BUS A) : BC -> RT            Msg No : 8 [3534818792]
Cmnd:   0828    (01-R-01-08)     RT1-SA01
Data:   0000  0047  0052  0007  0000  0000  0000  0000
Status: 0800
------------------------------------------------------------

(BUS A) : BC -> RT            Msg No : 9 [3534839536]
Cmnd:   0848    (01-R-02-08)     RT1-SA02
Data:   0000  0047  0052  0008  0000  0000  0000  0000
Status: 0800
------------------------------------------------------------


(BUS A) : BC -> RT            Msg No : 10 [3534860282]
Cmnd:   0868    (01-R-03-08)     RT1-SA03
Data:   0000  0047  0052  0009  0000  0000  0000  0000
Status: 0800
------------------------------------------------------------
W
(BUS A) : BC -> RT            Msg No : 11 [3534881047]
Cmnd:   0889    (01-R-04-09)     RT1-SA04
Data:   0000  0047  0052  000A  0000  0000  0000  0000
        0000
Status: 0800
------------------------------------------------------------
aiting until list processed
List processed.
Sleeping 20s
-------------  BC: START LIST EXECUTION -------------
Start RESULT list processing.
(BUS A) : RT -> BC            Msg No : 12 [5539904302]
Cmnd:   0C28    (01-T-01-08)     RT1-SA01
Status: 0800
Data:   0020  0021  0022  0023  0024  0025  0026  0027
------------------------------------------------------------

(BUS A) : RT -> BC            Msg No : 13 [5539925046]
Cmnd:   0C48    (01-T-02-08)     RT1-SA02
Status: 0800
Data:   0040  0041  0042  0043  0044  0045  0046  0047
------------------------------------------------------------


(BUS A) : RT -> BC            Msg No : 14 [5539945792]
Cmnd:   0C68    (01-T-03-08)     RT1-SA03
Status: 0800
Data:   0060  0061  0062  0063  0064  0065  0066  0067
------------------------------------------------------------
W
(BUS A) : RT -> BC            Msg No : 15 [5539966557]
Cmnd:   0C89    (01-T-04-09)     RT1-SA04
Status: 0800
Data:   0080  0081  0082  0083  0084  0085  0086  0087
        0088
------------------------------------------------------------
aiting until list processed
List processed.
Response to message 1: (len: 8, tsw1: 800, tsw2: 0)
  0x20 (.)0x21 (.)0x22 (.)0x23 (.)0x24 (.)0x25 (.)0x26 (.)0x27 (.)
Response to message 2: (len: 8, tsw1: 800, tsw2: 0)
  0x40 (.)0x41 (A)0x42 (B)0x43 (C)0x44 (D)0x45 (E)0x46 (F)0x47 (G)
Response to message 3: (len: 8, tsw1: 800, tsw2: 0)
  0x60 (.)0x61 (a)0x62 (b)0x63 (c)0x64 (d)0x65 (e)0x66 (f)0x67 (g)
Response to message 4: (len: 9, tsw1: 800, tsw2: 0)
  0x80 (.)0x81 (.)0x82 (.)0x83 (.)0x84 (.)0x85 (.)0x86 (.)0x87 (.)0x88 (.)
------------------------------------------------------
Sleeping 15s
-------------  BC: START LIST EXECUTION -------------
Start CMD list processing.
(BUS A) : BC -> RT            Msg No : 16 [7059818829]
Cmnd:   0828    (01-R-01-08)     RT1-SA01
...
...
```


```
        0000
Status: 0800
------------------------------------------------------------
aiting until list processed
List processed.
Sleeping 20s
-------------  BC: START LIST EXECUTION -------------
Start RESULT list processing.
(BUS A) : RT -> BC       Msg No : 4 [2014904315]
Cmnd:   0C28    (01-T-01-08)     RT1-SA01
Status: 0800
Data:   0020  0021  0022  0023  0024  0025  0026  0027
------------------------------------------------------------

(BUS A) : RT -> BC       Msg No : 5 [2014925059]
Cmnd:   0C48    (01-T-02-08)     RT1-SA02
Status: 0800
Data:   0040  0041  0042  0043  0044  0045  0046  0047
------------------------------------------------------------


(BUS A) : RT -> BC       Msg No : 6 [2014945805]
Cmnd:   0C68    (01-T-03-08)     RT1-SA03
Status: 0800
Data:   0060  0061  0062  0063  0064  0065  0066  0067
------------------------------------------------------------
W
(BUS A) : RT -> BC       Msg No : 7 [2014966570]
Cmnd:   0C89    (01-T-04-09)     RT1-SA04
Status: 0800
Data:   0080  0081  0082  0083  0084  0085  0086  0087
        0088
------------------------------------------------------------
aiting until list processed
List processed.
Response to message 1: (len: 8, tsw1: 800, tsw2: 0)
  0x20 (.)0x21 (.)0x22 (.)0x23 (.)0x24 (.)0x25 (.)0x26 (.)0x27 (.)
Response to message 2: (len: 8, tsw1: 800, tsw2: 0)
  0x40 (.)0x41 (A)0x42 (B)0x43 (C)0x44 (D)0x45 (E)0x46 (F)0x47 (G)
Response to message 3: (len: 8, tsw1: 800, tsw2: 0)
  0x60 (.)0x61 (a)0x62 (b)0x63 (c)0x64 (d)0x65 (e)0x66 (f)0x67 (g)
Response to message 4: (len: 9, tsw1: 800, tsw2: 0)
  0x80 (.)0x81 (.)0x82 (.)0x83 (.)0x84 (.)0x85 (.)0x86 (.)0x87 (.)0x88 (.)
------------------------------------------------------
Sleeping 15s
-------------  BC: START LIST EXECUTION -------------
Start CMD list processing.
(BUS A) : BC -> RT       Msg No : 8 [3534818792]
Cmnd:   0828    (01-R-01-08)     RT1-SA01
Data:   0000  0047  0052  0007  0000  0000  0000  0000
Status: 0800
------------------------------------------------------------

(BUS A) : BC -> RT       Msg No : 9 [3534839536]
Cmnd:   0848    (01-R-02-08)     RT1-SA02
Data:   0000  0047  0052  0008  0000  0000  0000  0000
Status: 0800
------------------------------------------------------------


(BUS A) : BC -> RT       Msg No : 10 [3534860282]
Cmnd:   0868    (01-R-03-08)     RT1-SA03
Data:   0000  0047  0052  0009  0000  0000  0000  0000
Status: 0800
------------------------------------------------------------
W
(BUS A) : BC -> RT       Msg No : 11 [3534881047]
Cmnd:   0889    (01-R-04-09)     RT1-SA04
Data:   0000  0047  0052  000A  0000  0000  0000  0000
        0000
Status: 0800
------------------------------------------------------------
aiting until list processed
List processed.
Sleeping 20s
-------------  BC: START LIST EXECUTION -------------
Start RESULT list processing.
(BUS A) : RT -> BC       Msg No : 12 [5539904302]
Cmnd:   0C28    (01-T-01-08)     RT1-SA01
Status: 0800
Data:   0020  0021  0022  0023  0024  0025  0026  0027
------------------------------------------------------------

(BUS A) : RT -> BC       Msg No : 13 [5539925046]
Cmnd:   0C48    (01-T-02-08)     RT1-SA02
Status: 0800
Data:   0040  0041  0042  0043  0044  0045  0046  0047
------------------------------------------------------------


(BUS A) : RT -> BC       Msg No : 14 [5539945792]
Cmnd:   0C68    (01-T-03-08)     RT1-SA03
Status: 0800
Data:   0060  0061  0062  0063  0064  0065  0066  0067
------------------------------------------------------------
W
(BUS A) : RT -> BC       Msg No : 15 [5539966557]
Cmnd:   0C89    (01-T-04-09)     RT1-SA04
Status: 0800
Data:   0080  0081  0082  0083  0084  0085  0086  0087
        0088
------------------------------------------------------------
aiting until list processed
List processed.
Response to message 1: (len: 8, tsw1: 800, tsw2: 0)
  0x20 (.)0x21 (.)0x22 (.)0x23 (.)0x24 (.)0x25 (.)0x26 (.)0x27 (.)
Response to message 2: (len: 8, tsw1: 800, tsw2: 0)
  0x40 (.)0x41 (A)0x42 (B)0x43 (C)0x44 (D)0x45 (E)0x46 (F)0x47 (G)
Response to message 3: (len: 8, tsw1: 800, tsw2: 0)
  0x60 (.)0x61 (a)0x62 (b)0x63 (c)0x64 (d)0x65 (e)0x66 (f)0x67 (g)
Response to message 4: (len: 9, tsw1: 800, tsw2: 0)
  0x80 (.)0x81 (.)0x82 (.)0x83 (.)0x84 (.)0x85 (.)0x86 (.)0x87 (.)0x88 (.)
------------------------------------------------------
Sleeping 15s
-------------  BC: START LIST EXECUTION -------------
Start CMD list processing.
(BUS A) : BC -> RT       Msg No : 16 [7059818829]
Cmnd:   0828    (01-R-01-08)     RT1-SA01
...
...
```

4.4 Loadable CAN Module

`laysim-GR712RC` supports the loadable CAN module by using '`-mod MODULE`' on start-up. Figure 4-6 shows the result of `rtems-occan0` with the loadable CAN module. The loadable CAN Node module can be loaded with '`-mod ./Loadable/loadable-CAN-Nodes/laysim_user_can_module1.so`' on start-up.

Figure 4-6. CAN Node Simulation using loadable CAN module

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS RCC-1.3.1]$ laysim-gr712rc-dbt-cli -r -core0 rtems-occan0.elf -mod ../Loadable/loadable-CAN-Nodes/laysim_user_can_module1.so
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ../Loadable/loadable-CAN-Nodes/laysim_user_can_module1.so


Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - rtems-occan0.elf
   - .text, addr: 0x40000000, size 226096 bytes
   - .rtemsroset, addr: 0x40037330, size 160 bytes
   - .rtemsstack, addr: 0x40037400, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x40039400, size 6080 bytes
   - .bss, addr: 0x4003ABC0, size 24736 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
occanlib_open: Opening driver /dev/occan0
occanlib_open: allocating memory 12
******** Starting Gaisler OCCAN sample ********
Starting task 2
Task2: Entering rx loop
Task2: waiting 1s
Starting task 1
Task1: Entring TX loop
Task1: sending 1 STD msg
Task1: sending 3 EXT msg
Task2: Got 3 messages
MSG: STD length: 4, id: 0x0
MSGDATA: 0x02 0xc4 0x4f 0xf2   ..O.
MSG[1]: EXT length: 4, id: 0xa
MSGDATA[1]: 0x02 0xc4 0x4f 0xf2   ..O.
MSG[2]: EXT length: 8, id: 0xa
MSGDATA[2]: 0xaa 0xbb 0x11 0x22 0x55 0x12 0xff 0x00   ....U...
Task2: Got 1 messages
----- GAISLER MESSAGE -----
MSG[3]: EXT length: 7, id: 0x200a
MSGDATA[3]: 0x47 0x61 0x69 0x73 0x6c 0x65 0x72   Gaisler
---------------------------
Task2: waiting 1s
Task1: sending 1 STD msg
Task1: sending 3 EXT msg
Task2: Got 3 messages
MSG[4]: STD length: 5, id: 0x0
MSGDATA[4]: 0x02 0xc4 0x4f 0xf2 0x23   ..O..
MSG[5]: EXT length: 5, id: 0xb
MSGDATA[5]: 0x02 0xc4 0x4f 0xf2 0x23   ..O..
MSG[6]: EXT length: 8, id: 0xb
MSGDATA[6]: 0xaa 0xbb 0x11 0x22 0x55 0x12 0xff 0x00   ....U...
Task2: Got 1 messages
----- GAISLER MESSAGE -----
MSG[7]: EXT length: 7, id: 0x200b
MSGDATA[7]: 0x47 0x61 0x69 0x73 0x6c 0x65 0x72   Gaisler
---------------------------
Task2: waiting 1s
Task1: sending 1 STD msg
Task1: sending 3 EXT msg
Task2: Got 3 messages
MSG[8]: STD length: 4, id: 0x0
MSGDATA[8]: 0x02 0xc4 0x4f 0xf2   ..O.
MSG[9]: EXT length: 4, id: 0xc
MSGDATA[9]: 0x02 0xc4 0x4f 0xf2   ..O.
MSG[10]: EXT length: 8, id: 0xc
MSGDATA[10]: 0xaa 0xbb 0x11 0x22 0x55 0x12 0xff 0x00   ....U...
Task2: Got 1 messages
----- GAISLER MESSAGE -----
MSG[11]: EXT length: 7, id: 0x200c
MSGDATA[11]: 0x47 0x61 0x69 0x73 0x6c 0x65 0x72   Gaisler
---------------------------
Task2: waiting 1s
Task1: sending 1 STD msg
Task1: sending 3 EXT msg
Task2: Got 3 messages
MSG[12]: STD length: 5, id: 0x0
MSGDATA[12]: 0x02 0xc4 0x4f 0xf2 0x23   ..O..
MSG[13]: EXT length: 5, id: 0xd
MSGDATA[13]: 0x02 0xc4 0x4f 0xf2 0x23   ..O..
MSG[14]: EXT length: 8, id: 0xd
MSGDATA[14]: 0xaa 0xbb 0x11 0x22 0x55 0x12 0xff 0x00   ....U...
Task2: Got 1 messages
----- GAISLER MESSAGE -----
MSG[15]: EXT length: 7, id: 0x200d
MSGDATA[15]: 0x47 0x61 0x69 0x73 0x6c 0x65 0x72   Gaisler
---------------------------
Task2: waiting 1s
Task1: sending 1 STD msg
Task1: sending 3 EXT msg
Task2: Got 3 messages
MSG[16]: STD length: 4, id: 0x0
MSGDATA[16]: 0x02 0xc4 0x4f 0xf2   ..O.
MSG[17]: EXT length: 4, id: 0xe
MSGDATA[17]: 0x02 0xc4 0x4f 0xf2   ..O.
MSG[18]: EXT length: 8, id: 0xe
MSGDATA[18]: 0xaa 0xbb 0x11 0x22 0x55 0x12 0xff 0x00   ....U...
Task2: Got 1 messages
----- GAISLER MESSAGE -----
MSG[19]: EXT length: 7, id: 0x200e
...
...
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\RCC-1.3.1> laysim-gr712rc-dbt-cli -r -core0 rtems-occan0.elf -mod ..\Loadable\loadable-CAN-Nodes\laysim_user_can_module1-dbt-cli.dll
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ..\Loadable\loadable-CAN-Nodes\laysim_user_can_module1-dbt-cli.dll

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - rtems-occan0.elf
   - .text, addr: 0x40000000, size 226096 bytes
   - .rtemsroset, addr: 0x40037330, size 160 bytes
   - .rtemsstack, addr: 0x40037400, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x40039400, size 6080 bytes
   - .bss, addr: 0x4003ABC0, size 24736 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
occanlib_open: Opening driver /dev/occan0
occanlib_open: allocating memory 12
******** Starting Gaisler OCCAN sample ********
Starting task 2
Task2: Entering rx loop
Task2: waiting 1s
Starting task 1
Task1: Entring TX loop
Task1: sending 1 STD msg
Task1: sending 3 EXT msg
Task2: Got 3 messages
MSG: STD length: 4, id: 0x0
MSGDATA: 0x02 0xc4 0x4f 0xf2   ..O.
MSG[1]: EXT length: 4, id: 0xa
MSGDATA[1]: 0x02 0xc4 0x4f 0xf2   ..O.
MSG[2]: EXT length: 8, id: 0xa
MSGDATA[2]: 0xaa 0xbb 0x11 0x22 0x55 0x12 0xff 0x00   ....U...
Task2: Got 1 messages
----- GAISLER MESSAGE -----
MSG[3]: EXT length: 7, id: 0x200a
MSGDATA[3]: 0x47 0x61 0x69 0x73 0x6c 0x65 0x72   Gaisler
---------------------------
Task2: waiting 1s
Task1: sending 1 STD msg
Task1: sending 3 EXT msg
Task2: Got 3 messages
MSG[4]: STD length: 5, id: 0x0
MSGDATA[4]: 0x02 0xc4 0x4f 0xf2 0x23   ..O..
MSG[5]: EXT length: 5, id: 0xb
MSGDATA[5]: 0x02 0xc4 0x4f 0xf2 0x23   ..O..
MSG[6]: EXT length: 8, id: 0xb
MSGDATA[6]: 0xaa 0xbb 0x11 0x22 0x55 0x12 0xff 0x00   ....U...
Task2: Got 1 messages
----- GAISLER MESSAGE -----
MSG[7]: EXT length: 7, id: 0x200b
MSGDATA[7]: 0x47 0x61 0x69 0x73 0x6c 0x65 0x72   Gaisler
---------------------------
Task2: waiting 1s
Task1: sending 1 STD msg
Task1: sending 3 EXT msg
Task2: Got 3 messages
MSG[8]: STD length: 4, id: 0x0
MSGDATA[8]: 0x02 0xc4 0x4f 0xf2   ..O.
MSG[9]: EXT length: 4, id: 0xc
MSGDATA[9]: 0x02 0xc4 0x4f 0xf2   ..O.
MSG[10]: EXT length: 8, id: 0xc
MSGDATA[10]: 0xaa 0xbb 0x11 0x22 0x55 0x12 0xff 0x00   ....U...
Task2: Got 1 messages
----- GAISLER MESSAGE -----
MSG[11]: EXT length: 7, id: 0x200c
MSGDATA[11]: 0x47 0x61 0x69 0x73 0x6c 0x65 0x72   Gaisler
---------------------------
Task2: waiting 1s
Task1: sending 1 STD msg
Task1: sending 3 EXT msg
Task2: Got 3 messages
MSG[12]: STD length: 5, id: 0x0
MSGDATA[12]: 0x02 0xc4 0x4f 0xf2 0x23   ..O..
MSG[13]: EXT length: 5, id: 0xd
MSGDATA[13]: 0x02 0xc4 0x4f 0xf2 0x23   ..O..
MSG[14]: EXT length: 8, id: 0xd
MSGDATA[14]: 0xaa 0xbb 0x11 0x22 0x55 0x12 0xff 0x00   ....U...
Task2: Got 1 messages
----- GAISLER MESSAGE -----
MSG[15]: EXT length: 7, id: 0x200d
MSGDATA[15]: 0x47 0x61 0x69 0x73 0x6c 0x65 0x72   Gaisler
---------------------------
Task2: waiting 1s
Task1: sending 1 STD msg
Task1: sending 3 EXT msg
Task2: Got 3 messages
MSG[16]: STD length: 4, id: 0x0
MSGDATA[16]: 0x02 0xc4 0x4f 0xf2   ..O.
MSG[17]: EXT length: 4, id: 0xe
MSGDATA[17]: 0x02 0xc4 0x4f 0xf2   ..O.
MSG[18]: EXT length: 8, id: 0xe
MSGDATA[18]: 0xaa 0xbb 0x11 0x22 0x55 0x12 0xff 0x00   ....U...
Task2: Got 1 messages
----- GAISLER MESSAGE -----
MSG[19]: EXT length: 7, id: 0x200e
..
..
```

4.5 Loadable I2C Module

`laysim-GR712RC` supports the loadable I2C module by using '`-mod MODULE`' on start-up. Figure 4-7 shows the result of `rtems-i2cmst` with the loadable I2C module. The loadable I2C module can be loaded with '`-mod ./Loadable/loadable-I2C-Modules/laysim_user_i2c_module.so`' on start-up.

Figure 4-7. I2C Simulation using loadable I2C module

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS RCC-1.3.1]$ laysim-gr712rc-dbt-cli -r -core0 rtems-i2cmst.elf -mod ../
Loadable/loadable-I2C-Modules/laysim_user_i2c_module.so
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ../Loadable/loadable-I2C-Modules/laysim_user_i2c_module.so


Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - rtems-i2cmst.elf
   - .text, addr: 0x40000000, size 205792 bytes
   - .rtemsroset, addr: 0x400323E0, size 176 bytes
   - .rtemsstack, addr: 0x400324C0, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x400344C0, size 5120 bytes
   - .bss, addr: 0x400358C0, size 22992 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
******** Starting Gaisler I2CMST test ********
Starting test task...
Reading from EEPROM using 'raw' driver..
Reading 100 bytes from EEPROM
Contents of first 100 bytes of EEPROM:
0x00:   0       1       2       3
0x04:   4       5       6       7
0x08:   8       9       a       b
0x0C:   c       d       e       f
0x10:   10      11      12      13
0x14:   14      15      16      17
0x18:   18      19      1a      1b
0x1C:   1c      1d      1e      1f
0x20:   20      21      22      23
0x24:   24      25      26      27
0x28:   28      29      2a      2b
0x2C:   2c      2d      2e      2f
0x30:   30      31      32      33
0x34:   34      35      36      37
0x38:   38      39      3a      3b
0x3C:   3c      3d      3e      3f
0x40:   40      41      42      43
0x44:   44      45      46      47
0x48:   48      49      4a      4b
0x4C:   4c      4d      4e      4f
0x50:   50      51      52      53
0x54:   54      55      56      57
0x58:   58      59      5a      5b
0x5C:   5c      5d      5e      5f
0x60:   60      61      62      63
Writing 16 bytes to EEPROM, starting at address 0x00: done..
Reading the first 20 bytes of the EEPROM:
0x00:   0       1       2       3
0x04:   4       5       6       7
0x08:   8       9       a       b
0x0C:   c       d       e       f
0x10:   10      11      12      13
Test completed, no more output will come from this test application
TEST OK
CORE#0 halts because of Error Mode [0.032486sec/41905284, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 41905284
   Instructions : 26430757
   CORE#0 performance (100MHz) : 63.07 MIPS
   Simulated time              : 419.05 ms

 Performance of laysim-gr712rc
   Real-time performance       : 1289.95 %
   laysim-gr712rc performance  : 813.60 MIPS
   Wall clock                  : 0.03 s
[layright@CentOS RCC-1.3.1]$
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\RCC-1.3.1> laysim-gr712rc-dbt-cli -r -core0 rtems-i2cmst.elf -mod
..\Loadable\loadable-I2C-Modules\laysim_user_i2c_module-dbt-cli.dll
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ..\Loadable\loadable-I2C-Modules\laysim_user_i2c_module-dbt
-cli.dll

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - rtems-i2cmst.elf
   - .text, addr: 0x40000000, size 205792 bytes
   - .rtemsroset, addr: 0x400323E0, size 176 bytes
   - .rtemsstack, addr: 0x400324C0, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x400344C0, size 5120 bytes
   - .bss, addr: 0x400358C0, size 22992 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
******** Starting Gaisler I2CMST test ********
Starting test task...
Reading from EEPROM using 'raw' driver..
Reading 100 bytes from EEPROM
Contents of first 100 bytes of EEPROM:
0x00:   0       1       2       3
0x04:   4       5       6       7
0x08:   8       9       a       b
0x0C:   c       d       e       f
0x10:   10      11      12      13
0x14:   14      15      16      17
0x18:   18      19      1a      1b
0x1C:   1c      1d      1e      1f
0x20:   20      21      22      23
0x24:   24      25      26      27
0x28:   28      29      2a      2b
0x2C:   2c      2d      2e      2f
0x30:   30      31      32      33
0x34:   34      35      36      37
0x38:   38      39      3a      3b
0x3C:   3c      3d      3e      3f
0x40:   40      41      42      43
0x44:   44      45      46      47
0x48:   48      49      4a      4b
0x4C:   4c      4d      4e      4f
0x50:   50      51      52      53
0x54:   54      55      56      57
0x58:   58      59      5a      5b
0x5C:   5c      5d      5e      5f
0x60:   60      61      62      63
Writing 16 bytes to EEPROM, starting at address 0x00: done..
Reading the first 20 bytes of the EEPROM:
0x00:   0       1       2       3
0x04:   4       5       6       7
0x08:   8       9       a       b
0x0C:   c       d       e       f
0x10:   10      11      12      13
Test completed, no more output will come from this test application
TEST OK
CORE#0 halts because of Error Mode [0.186903sec/41905284, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 41905284
   Instructions : 26430757
   CORE#0 performance (100MHz) : 63.07 MIPS
   Simulated time              : 419.05 ms

 Performance of laysim-gr712rc
   Real-time performance       : 224.21 %
   laysim-gr712rc performance  : 141.41 MIPS
   Wall clock                  : 0.19 s
PS C:\laysim-gr712rc\RCC-1.3.1>
```

4.6 Loadable SPI Module

`laysim-GR712RC` supports the loadable SPI module by using '`-mod MODULE`' on start-up. Figure 4-8 shows the result of `rtems-spi` with the loadable SPI module. The loadable SPI module can be loaded with '`-mod ./Loadable/loadable-SPI-Modules/laysim_user_spi_module.so`' on start-up.

Figure 4-8. SPI Simulation using loadable SPI module

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS RCC-1.3.1]$ laysim-gr712rc-dbt-cli -r -core0 rtems-spi.elf -mod ../Loa
dable/loadable-SPI-Modules/laysim_user_spi_module.so
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ../Loadable/loadable-SPI-Modules/laysim_user_spi_module.so
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\RCC-1.3.1> laysim-gr712rc-dbt-cli -r -core0 rtems-spi.elf -mod ..
\Loadable\loadable-SPI-Modules\laysim_user_spi_module-dbt-cli.dll
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ..\Loadable\loadable-SPI-Modules\laysim_user_spi_module-dbt
```

```
Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - rtems-spi.elf
   - .text, addr: 0x40000000, size 212672 bytes
   - .rtemsroset, addr: 0x40033EC0, size 176 bytes
   - .rtemsstack, addr: 0x40033F80, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x40035F80, size 5696 bytes
   - .bss, addr: 0x400375C0, size 34592 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
******** Initializing SPICTRL test ********
Registering SPI FLASH driver: driver registered successfully


READ 0x0
Read, bytes: 15 [0x0, 0x0, 0x0, 0x0]
0x00:   0       0       0       0
0x04:   1       0       0       0
0x08:   2       0       0       0
0x0C:   3       0       0

READ 0xf
Read, bytes: 7 [0x0, 0x4]
0x0F:   0       4       0       0
0x13:   0       5       0

READ 0x10000
Read, bytes: 4 [0x0, 0x40]
0x10000:        0       40      0       0


READ 0x1fffa
Read, bytes: 10 [0x0, 0x0]
0x1FFFA:        0       0       ff      7f
0x1FFFE:        0       0       0       80
0x20002:        0       0

WRITE 0x1fffc


WRITE 0x03
CORE#0 halts because of Error Mode [0.017627sec/19945309, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 19945309
   Instructions : 12595358
   CORE#0 performance (100MHz) : 63.15 MIPS
   Simulated time              : 199.45 ms

 Performance of laysim-gr712rc
   Real-time performance       : 1131.52 %
   laysim-gr712rc performance  : 714.55 MIPS
   Wall clock                  : 0.02 s
[layright@CentOS RCC-1.3.1]$
```

```
-cli.dll

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - rtems-spi.elf
   - .text, addr: 0x40000000, size 212672 bytes
   - .rtemsroset, addr: 0x40033EC0, size 176 bytes
   - .rtemsstack, addr: 0x40033F80, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x40035F80, size 5696 bytes
   - .bss, addr: 0x400375C0, size 34592 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
******** Initializing SPICTRL test ********
Registering SPI FLASH driver: driver registered successfully


READ 0x0
Read, bytes: 15 [0x0, 0x0, 0x0, 0x0]
0x00:   0       0       0       0
0x04:   1       0       0       0
0x08:   2       0       0       0
0x0C:   3       0       0

READ 0xf
Read, bytes: 7 [0x0, 0x4]
0x0F:   0       4       0       0
0x13:   0       5       0

READ 0x10000
Read, bytes: 4 [0x0, 0x40]
0x10000:        0       40      0       0


READ 0x1fffa
Read, bytes: 10 [0x0, 0x0]
0x1FFFA:        0       0       ff      7f
0x1FFFE:        0       0       0       80
0x20002:        0       0

WRITE 0x1fffc


WRITE 0x03
CORE#0 halts because of Error Mode [0.0761734sec/19945309, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 19945309
   Instructions : 12595358
   CORE#0 performance (100MHz) : 63.15 MIPS
   Simulated time              : 199.45 ms

 Performance of laysim-gr712rc
   Real-time performance       : 261.84 %
   laysim-gr712rc performance  : 165.35 MIPS
   Wall clock                  : 0.08 s
PS C:\laysim-gr712rc\RCC-1.3.1>
```

4.7 Loadable GPIO Module

`laysim-GR712RC` supports the loadable GPIO module by using '`-mod MODULE`' on start-up. Figure 4-9 shows the result of `rtems-gpio` with the loadable GPIO module. The loadable GPIO module can be loaded with '`-mod ./Loadable/loadable-GPIO-Modules/laysim_user_gpio_module.so`' on start- up.

Figure 4-9. GPIO Simulation using loadable GPIO module

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS RCC-1.3.1]$ laysim-gr712rc-dbt-cli -r -core0 rtems-gpio.elf -mod ../Loadable/loadable-GPIO-Modules/laysim_user_gpio_module.so
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ../Loadable/loadable-GPIO-Modules/laysim_user_gpio_module.so

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - rtems-gpio.elf
   - .text, addr: 0x40000000, size 208848 bytes
   - .rtemsroset, addr: 0x40032FD0, size 160 bytes
   - .rtemsstack, addr: 0x40033080, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x40035080, size 6352 bytes
   - .bss, addr: 0x40036980, size 15136 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
GRGPIO[0x80000a00] PORT[31]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[30]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[29]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[28]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[27]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[26]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[25]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[24]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[23]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[22]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[21]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[20]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[19]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[18]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[17]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[16]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[15]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[14]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[13]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[12]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[11]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\RCC-1.3.1> laysim-gr712rc-dbt-cli -r -core0 rtems-gpio.elf -mod ..\Loadable\loadable-GPIO-Modules\laysim_user_gpio_module-dbt-cli.dll
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ..\Loadable\loadable-GPIO-Modules\laysim_user_gpio_module-dbt-cli.dll

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - rtems-gpio.elf
   - .text, addr: 0x40000000, size 208848 bytes
   - .rtemsroset, addr: 0x40032FD0, size 160 bytes
   - .rtemsstack, addr: 0x40033080, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x40035080, size 6352 bytes
   - .bss, addr: 0x40036980, size 15136 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
GRGPIO[0x80000a00] PORT[31]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[30]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[29]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[28]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[27]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[26]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[25]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[24]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[23]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[22]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[21]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[20]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[19]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[18]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[17]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[16]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000a00] PORT[15]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[14]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[13]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[12]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[11]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
```

```
GRGPIO[0x80000a00] PORT[10]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[9]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[8]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[7]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[6]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[5]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[4]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[3]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[2]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[1]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[0]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[31]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[30]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[29]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[28]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[27]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[26]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[25]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[24]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[23]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[22]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[21]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[20]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[19]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[18]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[17]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[16]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[15]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[14]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[13]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[12]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[11]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[10]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[9]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[8]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[7]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[6]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[5]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[4]: IN/OUT/DIR: [1,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[3]: IN/OUT/DIR: [1,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[2]: IN/OUT/DIR: [1,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[1]: IN/OUT/DIR: [1,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[0]: IN/OUT/DIR: [1,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
Failed to disable IRQ on gpio port 4, disabling port while not enabled should fail. OK
Current Value on GPIO: 0
GRGPIO[0x80000900] PORT[4]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [1,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[4]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [1,0,0], BYPASS: 0
TEST END
CORE#0 halts because of Error Mode [0.120455sec/164374035, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 164374035
   Instructions : 103864227
   CORE#0 performance (100MHz) : 63.19 MIPS
   Simulated time              : 1.64 s

 Performance of laysim-gr712rc
   Real-time performance       : 1364.61 %
   laysim-gr712rc performance  : 862.27 MIPS
   Wall clock                  : 0.12 s
[layright@CentOS RCC-1.3.1]$
```

```
GRGPIO[0x80000a00] PORT[10]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[9]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[8]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[7]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[6]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[5]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[4]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[3]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[2]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[1]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000a00] PORT[0]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[31]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[30]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[29]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[28]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[27]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[26]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[25]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[24]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[23]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[22]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[21]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[20]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[19]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[18]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[17]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[16]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[15]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[14]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[13]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[12]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[11]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[10]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[9]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[8]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[7]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[6]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[5]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[4]: IN/OUT/DIR: [1,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[3]: IN/OUT/DIR: [1,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[2]: IN/OUT/DIR: [1,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[1]: IN/OUT/DIR: [1,0,0], MASK/POL/EDGE: [0,1,1], BYPASS: 0
GRGPIO[0x80000900] PORT[0]: IN/OUT/DIR: [1,0,0], MASK/POL/EDGE: [0,0,0], BYPASS: 0
Failed to disable IRQ on gpio port 4, disabling port while not enabled should fail. OK
Current Value on GPIO: 0
GRGPIO[0x80000900] PORT[4]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [1,0,0], BYPASS: 0
GRGPIO[0x80000900] PORT[4]: IN/OUT/DIR: [0,0,0], MASK/POL/EDGE: [1,0,0], BYPASS: 0
TEST END
CORE#0 halts because of Error Mode [0.640598sec/164374035, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 164374035
   Instructions : 103864227
   CORE#0 performance (100MHz) : 63.19 MIPS
   Simulated time              : 1.64 s

 Performance of laysim-gr712rc
   Real-time performance       : 256.59 %
   laysim-gr712rc performance  : 162.14 MIPS
   Wall clock                  : 0.64 s
PS C:\laysim-gr712rc\RCC-1.3.1>
```

4.8 Loadable UART Module

`laysim-GR712RC` supports the loadable UART module by using '`-mod MODULE`' on start-up. Figure 4-10 shows the result of `dhry` with the loadable UART module. The loadable UART module can be loaded with '`-mod ./Loadable/loadable-UART-Modules/laysim_user_uart_module.so`' on start-up.

Figure 4-10. UART Simulation using loadable UART module

laysim-gr712rc-dbt-cli in Linux

```
[layright@CentOS BCC-2.2.1]$ laysim-gr712rc-dbt-cli -r -core0 dhry.elf -mod ../Loadable
/loadable-UART-Modules/laysim_user_uart_module.so
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ../Loadable/loadable-UART-Modules/laysim_user_uart_module.s
o

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - dhry.elf
   - .text, addr: 0x40000000, size 60160 bytes
   - .rodata, addr: 0x4000EB00, size 2048 bytes
   - .data, addr: 0x4000F300, size 1568 bytes
   - .bss, addr: 0x4000F920, size 10688 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
Execution starts, 1000000 runs through Dhrystone
Total execution time:          5.8 s
Microseconds for one run through Dhrystone:    5.8
Dhrystones per Second:                      172711.6

Dhrystones MIPS      :                       98.3

CORE#0 halts because of Error Mode [0.407811sec/586232747, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 586232747
   Instructions : 361317674
   CORE#0 performance (100MHz) : 61.63 MIPS
   Simulated time              : 5.86 s

 Performance of laysim-gr712rc
   Real-time performance       : 1437.51 %
   laysim-gr712rc performance  : 885.99 MIPS
```

laysim-gr712rc-dbt-cli in Windows

```
PS C:\laysim-gr712rc\BCC-2.2.1> laysim-gr712rc-dbt-cli -r -core0 dhry.elf -mod ..\Loada
ble\loadable-UART-Modules\laysim_user_uart_module-dbt-cli.dll
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Loaded user loadable module ..\Loadable\loadable-UART-Modules\laysim_user_uart_module-d
bt-cli.dll

Multicore control : Time quantum (200)
CORE#0 RAM file is loaded - dhry.elf
   - .text, addr: 0x40000000, size 60160 bytes
   - .rodata, addr: 0x4000EB00, size 2048 bytes
   - .data, addr: 0x4000F300, size 1568 bytes
   - .bss, addr: 0x4000F920, size 10688 bytes (not loaded, just info.)
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
Execution starts, 1000000 runs through Dhrystone
Total execution time:          5.8 s
Microseconds for one run through Dhrystone:    5.8
Dhrystones per Second:                      172711.6

Dhrystones MIPS      :                       98.3

CORE#0 halts because of Error Mode [0.421874sec/586232747, 0cycles]

 Performance statistics for CORE#0
   Cycles       : 586232747
   Instructions : 361317674
   CORE#0 performance (100MHz) : 61.63 MIPS
   Simulated time              : 5.86 s

 Performance of laysim-gr712rc
   Real-time performance       : 1389.59 %
   laysim-gr712rc performance  : 856.46 MIPS
```

```
    Wall clock                  : 0.41 s
[layright@CentOS BCC-2.2.1]$
```

```
    Wall clock                  : 0.42 s
PS C:\laysim-gr712rc\BCC-2.2.1>
```

5. Shared Library

The professional version of `laysim-GR712RC` is provides as a stand-alone application as well as a set of libraries that can be integrated in an existing simulator. Detailed information can be found in `[laysim-GR712RC-006] laysim-GR712RC Shared Library - User's Manual`.

5.1 Shared Library Example

`app7-1.c` loads `rtems-i2cmst.elf` to CORE#0 and loads the loadable I2C module (`laysim_user_i2c_module.so`) and then it runs till 300000000 nsec. After execution, it saves all states of `laysim-GR712RC` and loadable I2C module to files. `app7-2.c` restores all states of `laysim-GR712RC` and loadable I2C module from files. And then it continues the execution. Figure 5-2 shows the result of `laysim-gr712rc-cli` and `laysim-gr712rc-dbt-cli`.

Figure 5-1. Source Code - app7-1.c & app7-2.c

```
001    #include <stdio.h>
002    #include "laysim_gr712rc_lib_defs.h"
003    #include "laysim_gr712rc_lib_prot.h"
004
005    int main (void)
006    {
007       INT8 result;
008
009       /* Step 1 : Initialize laysim-GR712RC environment */
010       result = laysim_gr712rc_lib_init (8192, 8192, 100, 38400);
011
012       /* Step 2 : Load rtems-i2cmst.elf to CORE#0 */
013       result = laysim_gr712rc_lib_load_images ("../rtems-i2cmst.elf", NULL, NULL);
014       if (result != 0)
015       {
016          /* Step 2-1 : Exit laysim-GR712RC */
017          laysim_gr712rc_lib_exit ();
018
019          /* Step 2-2 : Return */
020          return 0;
021       }
022
023       /* Step 3 : Load user I2C module */
024       #ifdef LINUX
025       result = laysim_gr712rc_lib_register_user_module ("../../Loadable/loadable-I2C-Modules/laysim_user_i2c_module.so");
026       #else
027       #if defined DBT
028       result = laysim_gr712rc_lib_register_user_module ("../../Loadable/loadable-I2C-Modules/laysim_user_i2c_module-dbt-cli-shared.dll");
029       #else
030       result = laysim_gr712rc_lib_register_user_module ("../../Loadable/loadable-I2C-Modules/laysim_user_i2c_module-cli-shared.dll");
031       #endif
032       #endif
033       if (result != 0)
034       {
035          /* Step 3-1 : Exit laysim-GR712RC */
036          laysim_gr712rc_lib_exit ();
037
038          /* Step 3-2 : Return */
039          return 0;
040       }
041
042       /* Step 4 : Run with delta 30000000 cycles (300000000nsec) */
043       result = laysim_gr712rc_lib_run_with_delta_cycles (30000000);
044
045       /* Step 5 : Save laysim-GR712RC state */
046       result = laysim_gr712rc_lib_save ("app7-1");
047
048       /* Step 6 : Exit laysim-GR712RC */
049       laysim_gr712rc_lib_exit ();
050
051       /* Step 7 : Return */
052       return 0;
053    }
```

```
001    #include <stdio.h>
002    #include "laysim_gr712rc_lib_defs.h"
003    #include "laysim_gr712rc_lib_prot.h"
004
005    int main (void)
006    {
007       INT8 result;
008
009       /* Step 1 : Initialize laysim-GR712RC environment */
010       result = laysim_gr712rc_lib_init (8192, 8192, 100, 38400);
011
012       /* Step 2 : Load user I2C module */
013       #ifdef LINUX
014       result = laysim_gr712rc_lib_register_user_module ("../../Loadable/loadable-I2C-Modules/laysim_user_i2c_module.so");
015       #else
016       #if defined DBT
017       result = laysim_gr712rc_lib_register_user_module ("../../Loadable/loadable-I2C-Modules/laysim_user_i2c_module-dbt-cli-shared.dll");
018       #else
019       result = laysim_gr712rc_lib_register_user_module ("../../Loadable/loadable-I2C-Modules/laysim_user_i2c_module-cli-shared.dll");
020       #endif
021       #endif
```

```
022       if (result != 0)
023       {
024          /* Step 2-1 : Exit laysim-GR712RC */
025          laysim_gr712rc_lib_exit ();
026
027          /* Step 2-2 : Return */
028          return 0;
029       }
030
031       /* Step 3 : Restore laysim-GR712RC state */
032       result = laysim_gr712rc_lib_restore ("app7-1");
033       if (result != 0)
034       {
035          /* Step 3-1 : Exit laysim-GR712RC */
036          laysim_gr712rc_lib_exit ();
037
038          /* Step 3-2 : Return */
039          return 0;
040       }
041
042       /* Step 4 : Run */
043       result = laysim_gr712rc_lib_run ();
044
045       /* Step 5 : Exit laysim-GR712RC */
046       laysim_gr712rc_lib_exit ();
047
048       /* Step 6 : Return */
049       return 0;
050    }
```

Figure 5-2. app7-1 and app7-2 Result

laysim-gr712rc-cli Result

```
[layright@CentOS laysim-gr712rc-cli]$ ./app7-1.exe
CORE#0 RAM file is loaded - ../rtems-i2cmst.elf
   - .text, addr: 0x40000000, size 205792 bytes
   - .rtemsroset, addr: 0x400323E0, size 176 bytes
   - .rtemsstack, addr: 0x400324C0, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x400344C0, size 5120 bytes
   - .bss, addr: 0x400358C0, size 22992 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
Loaded user loadable module ../../Loadable/loadable-I2C-Modules/laysim_user_i2c_module.
so
******** Starting Gaisler I2CMST test ********
Starting test task...
Reading from EEPROM using 'raw' driver..
Reading 100 bytes from EEPROM
Contents of first 100 bytes of EEPROM:
0x00:   0       1       2       3
0x04:   4       5       6       7
0x08:   8       9       a       b
0x0C:   c       d       e       f
0x10:   10      11      12      13
0x14:   14      15      16      17
0x18:   18      19      1a      1b
0x1C:   1c      1d      1e      1f
0x20:   20      21      22      23
0x24:   24      25      26      27
0x28:   28      29      2a      2b
0x2C:   2c      2d      2e      2f
0x30:   30      31      32      33
0x34:   34      35      36      37
0x38:   38      39      3a      3b
0x3C:   3c      3d      3e      3f
0x40:   40      41      42      43
0x44:   44      45      46      47
0x48:   48      49      4a      4b
0x4C:   4c      4d      4e      4f
0x50:   50      51      52      53
0x54:   54      55      56      57
0x58:   58      59      5a
 laysim-GR712RC state saved to : app7-1.v3.1.lay
 --> Total events : 1

 0001    0x7fa01e7b9700   0x00000000    30012263
 I2C Module saved to : app7-1.v3.1.lay-i2c

[layright@CentOS laysim-gr712rc-cli]$ ./app7-2.exe
Loaded user loadable module ../../Loadable/loadable-I2C-Modules/laysim_user_i2c_module.
so

 laysim-GR712RC state is restored from : app7-1.v3.1.lay
 --> Total events : 1

 0001    0x7fa01e7b9700   0x00000000    30012263
   - Detect RTEMS v5.1(SMP) with thread information

 I2C Module restored from : app7-1.v3.1.lay-i2c   5b
0x5C:   5c      5d      5e      5f
0x60:   60      61      62      63
Writing 16 bytes to EEPROM, starting at address 0x00: done..
Reading the first 20 bytes of the EEPROM:
0x00:   0       1       2       3
0x04:   4       5       6       7
0x08:   8       9       a       b
0x0C:   c       d       e       f
0x10:   10      11      12      13
Test completed, no more output will come from this test application
TEST OK
CORE#0 halts because of Error Mode [42526905cycles/0.451452sec]

[layright@CentOS laysim-gr712rc-cli]$
```

laysim-gr712rc-dbt-cli Result

```
[layright@CentOS laysim-gr712rc-dbt-cli]$ ./app7-1.exe
CORE#0 RAM file is loaded - ../rtems-i2cmst.elf
   - .text, addr: 0x40000000, size 205792 bytes
   - .rtemsroset, addr: 0x400323E0, size 176 bytes
   - .rtemsstack, addr: 0x400324C0, size 8192 bytes (not loaded, just info.)
   - .data, addr: 0x400344C0, size 5120 bytes
   - .bss, addr: 0x400358C0, size 22992 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000
Loaded user loadable module ../../Loadable/loadable-I2C-Modules/laysim_user_i2c_module.
so
******** Starting Gaisler I2CMST test ********
Starting test task...
Reading from EEPROM using 'raw' driver..
Reading 100 bytes from EEPROM
Contents of first 100 bytes of EEPROM:
0x00:   0       1       2       3
0x04:   4       5       6       7
0x08:   8       9       a       b
0x0C:   c       d       e       f
0x10:   10      11      12      13
0x14:   14      15      16      17
0x18:   18      19      1a      1b
0x1C:   1c      1d      1e      1f
0x20:   20      21      22      23
0x24:   24      25      26      27
0x28:   28      29      2a      2b
0x2C:   2c      2d      2e      2f
0x30:   30      31      32      33
0x34:   34      35      36      37
0x38:   38      39      3a      3b
0x3C:   3c      3d      3e      3f
0x40:   40      41      42      43
0x44:   44      45      46      47
0x48:   48      49      4a      4b
0x4C:   4c      4d      4e      4f
0x50:   50      51      52      53
0x54:   54      55      56      57
0x58:   58      59      5a      5b
0x5C:   5c
 laysim-GR712RC state saved to : app7-1.v3.1.dbt
 --> Total events : 1
 0001    0x7f94e58c9e60   0x00000000    30025080

 I2C Module saved to : app7-1.v3.1.dbt-i2c

[layright@CentOS laysim-gr712rc-dbt-cli]$ ./app7-2.exe
Loaded user loadable module ../../Loadable/loadable-I2C-Modules/laysim_user_i2c_module.
so
 laysim-GR712RC state is restored from : app7-1.v3.1.dbt
 --> Total events : 1
 0001    0x7f94e58c9e60   0x00000000    30025080

   - Detect RTEMS v5.1(SMP) with thread information

 I2C Module restored from : app7-1.v3.1.dbt-i2c
 5d      5e      5f
0x60:   60      61      62      63
Writing 16 bytes to EEPROM, starting at address 0x00: done..
Reading the first 20 bytes of the EEPROM:
0x00:   0       1       2       3
0x04:   4       5       6       7
0x08:   8       9       a       b
0x0C:   c       d       e       f
0x10:   10      11      12      13
Test completed, no more output will come from this test application
TEST OK
CORE#0 halts because of Error Mode [0.008858sec/41905285, 0cycles]

[layright@CentOS laysim-gr712rc-dbt-cli]$
```

6. GDB Operation

The CLI version of `laysim-GR712RC` supports debugging with the GDB. When attached to the GDB, `laysim-GR712RC` acts as a remote GDB target. Applications are loaded and debugged through the GDB. To initiate GDB communications, start the CLI version of `laysim-GR712RC` with the -gdb start-up option or use the gdb command in the control console. Detailed information can be found in `[laysim-GR712RC-003] Debugging with GDB on laysim-GR712RC`.

6.1 Using Extended-Remote Protocol

The CLI version of `laysim-GR712RC` supports the extended-remote protocol of GDB, so it is possible to run RTEMS 5.1 SMP examples using 'run' command in GDB.

Figure 6-1. Debugging with GDB #1 on laysim-GR712RC

laysim-gr712rc-cli Result

```
[layright@CentOS smptests]$ laysim-gr712rc-cli -gdb
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

--> No CORE#0 RAM file is loaded

 GDB interface: using port 1234
 Starting GDB server. Use Ctrl-C to stop waiting for connection
 GDB connected at IP: 127.0.0.1 and port: 45424


*** BEGIN OF TEST SMP 9 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
 CPU 1 start task TA1
 kill 10 clock ticks
-------------------------------------------------------------------------------
                          CPU USAGE BY THREAD
------------+-----------------------------------------+---------------+---------
 ID         | NAME                                    | SECONDS       | PERCENT
------------+-----------------------------------------+---------------+---------
 0x09010001 | IDLE                                    |     1.171035  |  91.028
 0x09010002 | IDLE                                    |     1.000131  |  76.363
 0x0a010001 | UI1                                     |     0.332825  |  24.968
 0x0a010002 | TA1                                     |     0.185180  |  13.654
------------+-----------------------------------------+---------------+---------
 TIME SINCE LAST CPU USAGE RESET IN SECONDS:                          1.356217
-------------------------------------------------------------------------------

*** END OF TEST SMP 9 ***


CORE#1 halts because of Error Mode [64222487cycles/0.580045sec]
```

```
[layright@CentOS smptests]$ sparc-gaisler-elf-gdb smp09.exe
GNU gdb (GDB) 8.2.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "--host=x86_64-pc-linux-gnu --target=sparc-gaisler-elf".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from smp09.exe...done.
(gdb) tar extended-remote localhost:1234
Remote debugging using localhost:1234
0x00000000 in ?? ()
(gdb) load
Loading section .text, size 0x1cdb0 lma 0x40000000
Loading section .rtemsroset, size 0x90 lma 0x4001cdb0
Loading section .data, size 0x530 lma 0x40024e40
Start address 0x40000000, load size 119664
Transfer rate: 116859 KB/sec, 7039 bytes/write.
(gdb) b Init
Breakpoint 1 at 0x400012b4: file ../../../../../../rtems/c/src/../../testsuites/smptest
s/smp09/init.c, line 42.
(gdb) monitor gdb postload
(gdb) cont
Continuing.
[Switching to Thread 2]

Thread 2 hit Breakpoint 1, Init (argument=1073910656)
    at ../../../../../../rtems/c/src/../../testsuites/smptests/smp09/init.c:42
42          {
(gdb) info thread
  Id   Target Id                  Frame
  1    Thread 1 (CORE#0 [running]) 0x4000e034 in _User_extensions_Iterate (
    arg=arg@entry=0x40026fc8 <_Thread_Objects>,
    visitor=0x4000de30 <_User_extensions_Thread_begin_visitor>,
    direction=direction@entry=CHAIN_ITERATOR_FORWARD)
    at ../../../../../rtems/c/src/../../cpukit/score/src/userextiterate.c:217
* 2    Thread 2 (CORE#1 [running]) Init (argument=1073910656)
    at ../../../../../../rtems/c/src/../../testsuites/smptests/smp09/init.c:42
(gdb) thread 1
[Switching to thread 1 (Thread 1)]
#0  0x4000e034 in _User_extensions_Iterate (
    arg=arg@entry=0x40026fc8 <_Thread_Objects>,
    visitor=0x4000de30 <_User_extensions_Thread_begin_visitor>,
    direction=direction@entry=CHAIN_ITERATOR_FORWARD)
    at ../../../../../rtems/c/src/../../cpukit/score/src/userextiterate.c:217
217             if ( direction == CHAIN_ITERATOR_BACKWARD ) {
(gdb) bt
#0  0x4000e034 in _User_extensions_Iterate (
    arg=arg@entry=0x40026fc8 <_Thread_Objects>,
    visitor=0x4000de30 <_User_extensions_Thread_begin_visitor>,
    direction=direction@entry=CHAIN_ITERATOR_FORWARD)
    at ../../../../../rtems/c/src/../../cpukit/score/src/userextiterate.c:217
#1  0x40009e90 in _User_extensions_Thread_begin (
    executing=0x40026fc8 <_Thread_Objects>)
    at /home/layright/RTEMS-dev/RTEMS5-builder/rtems/cpukit/include/rtems/score/userext
impl.h:353
#2  _Thread_Handler ()
    at ../../../../../rtems/c/src/../../cpukit/score/src/threadhandler.c:130
#3  0x40009dcc in _Thread_Handler ()
    at ../../../../../rtems/c/src/../../cpukit/score/src/threadhandler.c:83
(gdb) thread 2
[Switching to thread 2 (Thread 2)]
#0  Init (argument=1073910656)
    at ../../../../../../rtems/c/src/../../testsuites/smptests/smp09/init.c:42
42          {
(gdb) bt
#0  Init (argument=1073910656)
    at ../../../../../../rtems/c/src/../../testsuites/smptests/smp09/init.c:42
#1  0x4000ac20 in _Thread_Entry_adaptor_numeric (
    executing=0x40025580 <_RTEMS_tasks_Objects>)
    at ../../../../../rtems/c/src/../../cpukit/score/src/threadentryadaptornumeric.c:25
#2  0x40009eb4 in _Thread_Handler ()
    at ../../../../../rtems/c/src/../../cpukit/score/src/threadhandler.c:139
#3  0x40009dcc in _Thread_Handler ()
    at ../../../../../rtems/c/src/../../cpukit/score/src/threadhandler.c:83
```

```
(gdb) cont
Continuing.

Thread 2 received signal SIGTERM, Terminated.
syscall ()
    at ../../../../../rtems/c/src/../../cpukit/score/cpu/sparc/syscall.S:44
44                          ta          0
! syscall 1, halt with %g1,%g2,%g3 info
(gdb) info thread
  Id   Target Id                    Frame
  1    Thread 1 (CORE#0 [running]) dont_do_the_window ()
    at ../../../../../rtems/c/src/../../cpukit/score/cpu/sparc/cpu_asm.S:455
* 2    Thread 2 (CORE#1 [error  ]) syscall ()
    at ../../../../../rtems/c/src/../../cpukit/score/cpu/sparc/syscall.S:44
(gdb) thread 1
[Switching to thread 1 (Thread 1)]
#0  dont_do_the_window ()
    at ../../../../../rtems/c/src/../../cpukit/score/cpu/sparc/cpu_asm.S:455
455                     SPARC_LEON3FT_B2BST_NOP
(gdb) bt
#0  dont_do_the_window ()
    at ../../../../../rtems/c/src/../../cpukit/score/cpu/sparc/cpu_asm.S:455
#1  0x400175d8 in pil_fixed ()
    at ../../../../../rtems/c/src/../../cpukit/score/cpu/sparc/cpu_asm.S:592
Backtrace stopped: previous frame identical to this frame (corrupt stack?)
(gdb) thread 2
[Switching to thread 2 (Thread 2)]
#0  syscall ()
    at ../../../../../rtems/c/src/../../cpukit/score/cpu/sparc/syscall.S:44
44                          ta          0
! syscall 1, halt with %g1,%g2,%g3 info
(gdb) bt
#0  syscall ()
    at ../../../../../rtems/c/src/../../cpukit/score/cpu/sparc/syscall.S:44
#1  0x4000de9c in _User_extensions_Fatal_visitor (executing=0x5, arg=0x0,
    callouts=0x400250a0 <_User_extensions_List+4>)
    at ../../../../../rtems/c/src/../../cpukit/score/src/userextiterate.c:131
#2  0x4001af14 in _CPU_Fatal_halt (source=source@entry=5, error=error@entry=0)
    at ../../../../../../../../rtems/c/src/lib/libbsp/sparc/leon3/../../../../../../bsp
s/sparc/leon3/start/bsp_fatal_halt.c:33
#3  0x4000e198 in _Terminate (
    the_source=the_source@entry=RTEMS_FATAL_SOURCE_EXIT,
    the_error=the_error@entry=0)
    at ../../../../../rtems/c/src/../../cpukit/score/src/interr.c:45
#4  0x4001b0e4 in rtems_shutdown_executive (result=result@entry=0)
    at ../../../../../rtems/c/src/../../cpukit/sapi/src/exshutdown.c:21
#5  0x4001a610 in rtems_test_exit (status=status@entry=0)
    at ../../../../../rtems/c/src/../../cpukit/libtest/testbeginend.c:86
#6  0x40001478 in Init (argument=<optimized out>)
    at ../../../../../../rtems/c/src/../../testsuites/smptests/smp09/init.c:83
#7  0x4000ac20 in _Thread_Entry_adaptor_numeric (
    executing=0x40025580 <_RTEMS_tasks_Objects>)
    at ../../../../../rtems/c/src/../../cpukit/score/src/threadentryadaptornumeric.c:25
#8  0x40009eb4 in _Thread_Handler ()
    at ../../../../../rtems/c/src/../../cpukit/score/src/threadhandler.c:139
#9  0x40009dcc in _Thread_Handler ()
    at ../../../../../rtems/c/src/../../cpukit/score/src/threadhandler.c:83
(gdb)
```

laysim-gr712rc-dbt-cli Result

```
[layright@CentOS smptests]$ laysim-gr712rc-dbt-cli -gdb
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
--> No CORE#0 RAM file is loaded
 GDB interface: using port 1234
 Starting GDB server. Use Ctrl-C to stop waiting for connection

 GDB connected at IP: 127.0.0.1 and port: 45426


*** BEGIN OF TEST SMP 9 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
 CPU 1 start task TA1
 kill 10 clock ticks
-------------------------------------------------------------------------------
                            CPU USAGE BY THREAD
------------+-----------------------------------+---------------+---------
 ID         | NAME                              | SECONDS       | PERCENT
------------+-----------------------------------+---------------+---------
 0x09010001 | IDLE                              |      1.200689 |  91.229
 0x09010002 | IDLE                              |      1.000049 |  74.665
 0x0a010001 | UI1                               |      0.362585 |  26.609
 0x0a010002 | TA1                               |      0.185203 |  13.363
------------+-----------------------------------+---------------+---------
 TIME SINCE LAST CPU USAGE RESET IN SECONDS:                     1.385895
-------------------------------------------------------------------------------

*** END OF TEST SMP 9 ***

CORE#1 halts because of Error Mode [0.051735sec/66502098, 66501899cycles]
```

```
[layright@CentOS smptests]$ sparc-gaisler-elf-gdb smp09.exe
GNU gdb (GDB) 8.2.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "--host=x86_64-pc-linux-gnu --target=sparc-gaisler-elf".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from smp09.exe...done.
(gdb) tar extended-remote localhost:1234
Remote debugging using localhost:1234
0x00000000 in ?? ()
(gdb) load
Loading section .text, size 0x1cdb0 lma 0x40000000
Loading section .rtemsroset, size 0x90 lma 0x4001cdb0
Loading section .data, size 0x530 lma 0x40024e40
Start address 0x40000000, load size 119664
Transfer rate: 116859 KB/sec, 7039 bytes/write.
(gdb) b Init
Breakpoint 1 at 0x400012b4: file ../../../../../../rtems/c/src/../../testsuites/smptest
s/smp09/init.c, line 42.
(gdb) monitor gdb postload
(gdb) cont
Continuing.
[Switching to Thread 2]

Thread 2 hit Breakpoint 1, Init (argument=1073910656)
    at ../../../../../../rtems/c/src/../../testsuites/smptests/smp09/init.c:42
42          {
(gdb) info thread
  Id   Target Id                    Frame
  1    Thread 1 (CORE#0 [running]) leon3_power_down_loop ()
    at ../../../../../../../../rtems/c/src/lib/libbsp/sparc/leon3/../../../../../../bsp
s/sparc/leon3/start/bspidle.S:26
* 2    Thread 2 (CORE#1 [running]) Init (argument=1073910656)
    at ../../../../../../rtems/c/src/../../testsuites/smptests/smp09/init.c:42
(gdb) thread 1
[Switching to thread 1 (Thread 1)]
#0  leon3_power_down_loop ()
    at ../../../../../../../../rtems/c/src/lib/libbsp/sparc/leon3/../../../../../../bsp
s/sparc/leon3/start/bspidle.S:26
26                      lda   [%sp] 1, %g0            ! Needed for UT699 and GR712
(gdb) bt
#0  leon3_power_down_loop ()
    at ../../../../../../../../rtems/c/src/lib/libbsp/sparc/leon3/../../../../../../bsp
s/sparc/leon3/start/bspidle.S:26
#1  0x400198bc in _Thread_Entry_adaptor_idle (
    executing=0x40026fc8 <_Thread_Objects>)
    at ../../../../../rtems/c/src/../../cpukit/score/src/threadentryadaptoridle.c:25
#2  0x40009eb4 in _Thread_Handler ()
    at ../../../../../rtems/c/src/../../cpukit/score/src/threadhandler.c:139
```

```
#3  0x40009dcc in _Thread_Handler ()
    at ../../../../../rtems/c/src/../../cpukit/score/src/threadhandler.c:83
(gdb) thread 2
[Switching to thread 2 (Thread 2)]
#0  Init (argument=1073910656)
    at ../../../../../../rtems/c/src/../../testsuites/smptests/smp09/init.c:42
42          {
(gdb) bt
#0  Init (argument=1073910656)
    at ../../../../../../rtems/c/src/../../testsuites/smptests/smp09/init.c:42
#1  0x4000ac20 in _Thread_Entry_adaptor_numeric (
    executing=0x40025580 <_RTEMS_tasks_Objects>)
    at ../../../../../rtems/c/src/../../cpukit/score/src/threadentryadaptornumeric.c:25
#2  0x40009eb4 in _Thread_Handler ()
    at ../../../../../rtems/c/src/../../cpukit/score/src/threadhandler.c:139
#3  0x40009dcc in _Thread_Handler ()
    at ../../../../../rtems/c/src/../../cpukit/score/src/threadhandler.c:83
(gdb) cont
Continuing.

Thread 2 received signal SIGTERM, Terminated.
syscall ()
    at ../../../../../rtems/c/src/../../cpukit/score/cpu/sparc/syscall.S:44
44                          ta          0
! syscall 1, halt with %g1,%g2,%g3 info
(gdb) info thread
  Id   Target Id                Frame
  1    Thread 1 (CORE#0 [running]) bsp_inter_processor_interrupt (vector=30)
    at ../../../../../../../../rtems/c/src/lib/libbsp/sparc/leon3/../../../../../../bsp
s/sparc/leon3/start/bspsmp.c:34
* 2    Thread 2 (CORE#1 [error  ]) syscall ()
    at ../../../../../rtems/c/src/../../cpukit/score/cpu/sparc/syscall.S:44
(gdb) thread 1
[Switching to thread 1 (Thread 1)]
#0  bsp_inter_processor_interrupt (vector=30)
    at ../../../../../../../../rtems/c/src/lib/libbsp/sparc/leon3/../../../../../../bsp
s/sparc/leon3/start/bspsmp.c:34
34          {
(gdb) bt
#0  bsp_inter_processor_interrupt (vector=30)
    at ../../../../../../../../rtems/c/src/lib/libbsp/sparc/leon3/../../../../../../bsp
s/sparc/leon3/start/bspsmp.c:34
#1  0x400175d8 in pil_fixed ()
    at ../../../../../rtems/c/src/../../cpukit/score/cpu/sparc/cpu_asm.S:592
Backtrace stopped: previous frame identical to this frame (corrupt stack?)
(gdb) thread 2
[Switching to thread 2 (Thread 2)]
#0  syscall ()
    at ../../../../../rtems/c/src/../../cpukit/score/cpu/sparc/syscall.S:44
44                          ta          0
! syscall 1, halt with %g1,%g2,%g3 info
(gdb) bt
#0  syscall ()
    at ../../../../../rtems/c/src/../../cpukit/score/cpu/sparc/syscall.S:44
#1  0x4000de9c in _User_extensions_Fatal_visitor (executing=0x5, arg=0x0,
    callouts=0x400250a0 <_User_extensions_List+4>)
    at ../../../../../rtems/c/src/../../cpukit/score/src/userextiterate.c:131
#2  0x4001af14 in _CPU_Fatal_halt (source=source@entry=5, error=error@entry=0)
    at ../../../../../../../../rtems/c/src/lib/libbsp/sparc/leon3/../../../../../../bsp
s/sparc/leon3/start/bsp_fatal_halt.c:33
#3  0x4000e198 in _Terminate (
    the_source=the_source@entry=RTEMS_FATAL_SOURCE_EXIT,
    the_error=the_error@entry=0)
    at ../../../../../rtems/c/src/../../cpukit/score/src/interr.c:45
#4  0x4001b0e4 in rtems_shutdown_executive (result=result@entry=0)
    at ../../../../../rtems/c/src/../../cpukit/sapi/src/exshutdown.c:21
#5  0x4001a610 in rtems_test_exit (status=status@entry=0)
    at ../../../../../rtems/c/src/../../cpukit/libtest/testbeginend.c:86
#6  0x40001478 in Init (argument=<optimized out>)
    at ../../../../../../rtems/c/src/../../testsuites/smptests/smp09/init.c:83
#7  0x4000ac20 in _Thread_Entry_adaptor_numeric (
    executing=0x40025580 <_RTEMS_tasks_Objects>)
    at ../../../../../rtems/c/src/../../cpukit/score/src/threadentryadaptornumeric.c:25
#8  0x40009eb4 in _Thread_Handler ()
    at ../../../../../rtems/c/src/../../cpukit/score/src/threadhandler.c:139
#9  0x40009dcc in _Thread_Handler ()
    at ../../../../../rtems/c/src/../../cpukit/score/src/threadhandler.c:83
(gdb)
```

6.2 Using Remote Protocol

The CLI version of `laysim-GR712RC` supports the remote protocol of GDB, so it is required to set the SMP configuration using 'monitor gdb postload' after loading RTEMS 5.1 SMP examples. The remote protocol of GDB doesn't support 'run' command, so 'cont' command should be used.

Figure 6-2. Debugging with GDB #2 on laysim-GR712RC

laysim-gr712rc-cli Result

```
[layright@CentOS smptests]$ laysim-gr712rc-cli -gdb
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

--> No CORE#0 RAM file is loaded

 GDB interface: using port 1234
 Starting GDB server. Use Ctrl-C to stop waiting for connection
 GDB connected at IP: 127.0.0.1 and port: 45440


*** BEGIN OF TEST SMP 9 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
 CPU 1 start task TA1
```

```
[layright@CentOS smptests]$ sparc-gaisler-elf-gdb smp09.exe
GNU gdb (GDB) 8.2.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "--host=x86_64-pc-linux-gnu --target=sparc-gaisler-elf".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
```

```
 kill 10 clock ticks
-------------------------------------------------------------------------------
                            CPU USAGE BY THREAD
------------+---------------------------------------+--------------+---------
 ID         | NAME                                  | SECONDS      | PERCENT
------------+---------------------------------------+--------------+---------
 0x09010001 | IDLE                                  |     1.171035 |  91.028
 0x09010002 | IDLE                                  |     1.000131 |  76.363
 0x0a010001 | UI1                                   |     0.332825 |  24.968
 0x0a010002 | TA1                                   |     0.185180 |  13.654
------------+---------------------------------------+--------------+---------
 TIME SINCE LAST CPU USAGE RESET IN SECONDS:                       1.356217
-------------------------------------------------------------------------------

*** END OF TEST SMP 9 ***


CORE#1 halts because of Error Mode [64222487cycles/0.570294sec]

 GDB: disconnected

 GDB interface: using port 1234
 Starting GDB server. Use Ctrl-C to stop waiting for connection
```

```
Reading symbols from smp09.exe...done.
(gdb) tar remote localhost:1234
Remote debugging using localhost:1234
0x00000000 in ?? ()
(gdb) load
Loading section .text, size 0x1cdb0 lma 0x40000000
Loading section .rtemsroset, size 0x90 lma 0x4001cdb0
Loading section .data, size 0x530 lma 0x40024e40
Start address 0x40000000, load size 119664
Transfer rate: 116859 KB/sec, 7039 bytes/write.
(gdb) monitor gdb postload
(gdb) b Init
Breakpoint 1 at 0x400012b4: file ../../../../../../rtems/c/src/../../testsuites/smptest
s/smp09/init.c, line 42.
(gdb) cont
Continuing.
[Switching to Thread 2]

Thread 2 hit Breakpoint 1, Init (argument=1073910656)
    at ../../../../../../rtems/c/src/../../testsuites/smptests/smp09/init.c:42
42            {
(gdb) step
49              TEST_BEGIN();
(gdb) step
rtems_test_begin (name=name@entry=0x4001be90 <rtems_test_name> "SMP 9",
    state=state@entry=RTEMS_TEST_STATE_PASS)
    at ../../../../../../rtems/c/src/../../cpukit/libtest/testbeginend.c:40
40              return rtems_printf(
(gdb) cont
Continuing.
[Inferior 1 (Remote target) exited normally]
(gdb)
```

laysim-gr712rc-dbt-cli Result

```
[layright@CentOS smptests]$ laysim-gr712rc-dbt-cli -gdb
laysim for GR712RC Dual Core Processor (DBT) CLI v3.1 by layright

Multicore control : Time quantum (200)
--> No CORE#0 RAM file is loaded
 GDB interface: using port 1234
 Starting GDB server. Use Ctrl-C to stop waiting for connection

 GDB connected at IP: 127.0.0.1 and port: 45442


*** BEGIN OF TEST SMP 9 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
 CPU 1 start task TA1
 kill 10 clock ticks
-------------------------------------------------------------------------------
                            CPU USAGE BY THREAD
------------+---------------------------------------+--------------+---------
 ID         | NAME                                  | SECONDS      | PERCENT
------------+---------------------------------------+--------------+---------
 0x09010001 | IDLE                                  |     1.200689 |  91.229
 0x09010002 | IDLE                                  |     1.000049 |  74.665
 0x0a010001 | UI1                                   |     0.362585 |  26.609
 0x0a010002 | TA1                                   |     0.185203 |  13.363
------------+---------------------------------------+--------------+---------
 TIME SINCE LAST CPU USAGE RESET IN SECONDS:                       1.385895
-------------------------------------------------------------------------------

*** END OF TEST SMP 9 ***

CORE#1 halts because of Error Mode [0.048163sec/66502098, 66501899cycles]

 GDB: disconnected
 GDB interface: using port 1234
 Starting GDB server. Use Ctrl-C to stop waiting for connection
```

```
[layright@CentOS smptests]$ sparc-gaisler-elf-gdb smp09.exe
GNU gdb (GDB) 8.2.1
Copyright (C) 2018 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "--host=x86_64-pc-linux-gnu --target=sparc-gaisler-elf".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from smp09.exe...done.
(gdb) tar remote localhost:1234
Remote debugging using localhost:1234
0x00000000 in ?? ()
(gdb) load
Loading section .text, size 0x1cdb0 lma 0x40000000
Loading section .rtemsroset, size 0x90 lma 0x4001cdb0
Loading section .data, size 0x530 lma 0x40024e40
Start address 0x40000000, load size 119664
Transfer rate: 116859 KB/sec, 7039 bytes/write.
(gdb) monitor gdb postload
(gdb) b Init
Breakpoint 1 at 0x400012b4: file ../../../../../../rtems/c/src/../../testsuites/smptest
s/smp09/init.c, line 42.
(gdb) cont
Continuing.
[Switching to Thread 2]

Thread 2 hit Breakpoint 1, Init (argument=1073910656)
    at ../../../../../../rtems/c/src/../../testsuites/smptests/smp09/init.c:42
42            {
(gdb) step
49              TEST_BEGIN();
(gdb) step
rtems_test_begin (name=name@entry=0x4001be90 <rtems_test_name> "SMP 9",
    state=state@entry=RTEMS_TEST_STATE_PASS)
    at ../../../../../../rtems/c/src/../../cpukit/libtest/testbeginend.c:40
40              return rtems_printf(
(gdb) cont
Continuing.
[Inferior 1 (Remote target) exited normally]
(gdb)
```

6.3 Using Eclipse

The CLI version of `laysim-GR712RC` can be connected to a GDB front-end such as DDD or Eclipse. Eclipse supports only the remote protocol of GDB, '`monitor gdb reset`' and '`monitor gdb postload`' should be added in Debug Configurations of Eclipse as shown in Figure 6-3.

Figure 6-3. Debug Configurations in Eclipse

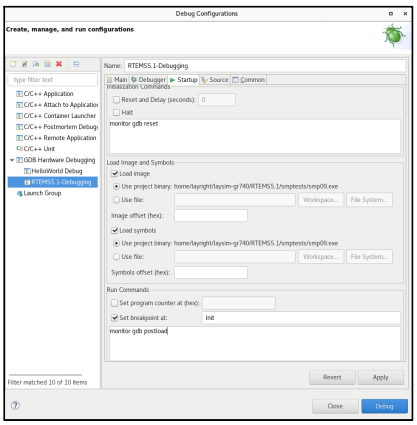
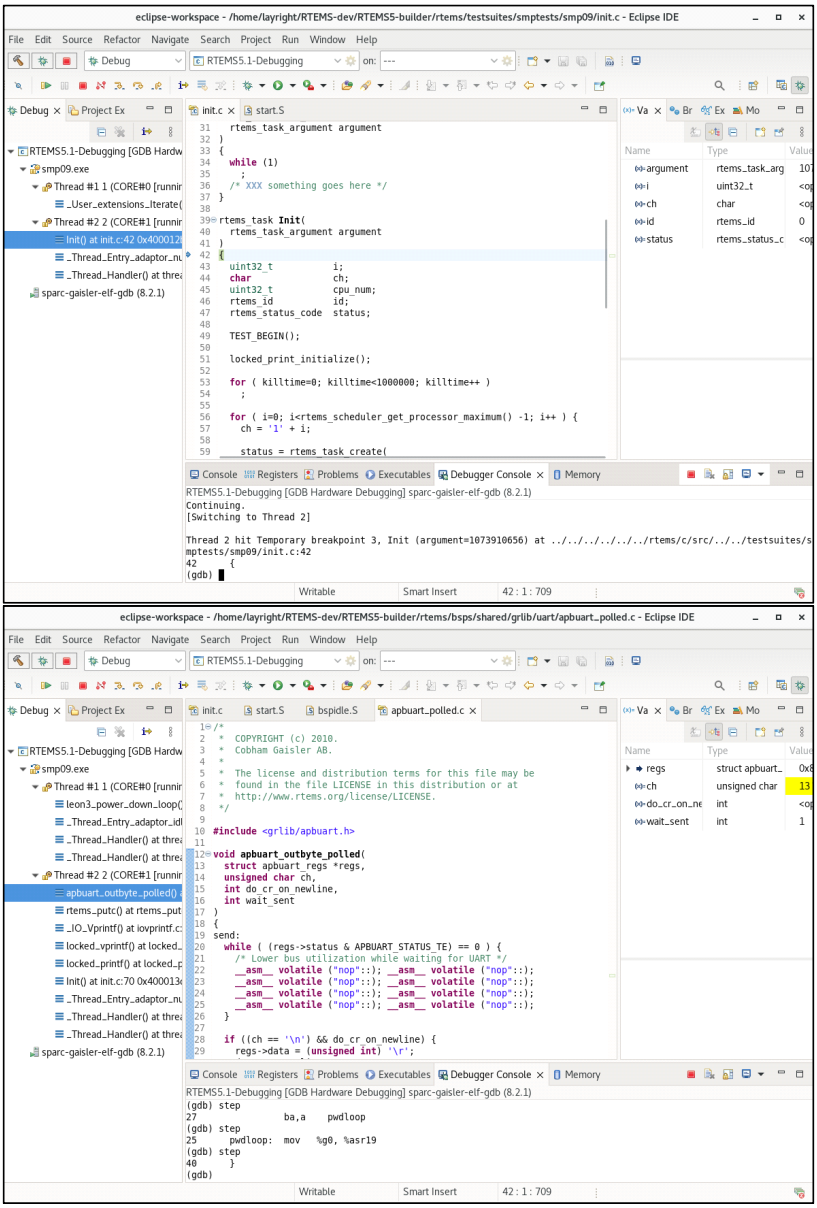


Figure 6-4. Debugging with Eclipse on laysim-GR712RC



7. Code Coverage

`laysim-gr712rc-cli` only supports for the code coverage in the same format as `TSIM3-LEON3`. '`-cov`' start-up option should be enabled for the code coverage. Detailed information can be found in `[laysim-GR712RC-004] Code coverage operation on laysim-GR712RC`.

Figure 7-1 shows the result of `smp09.exe` with code coverage. After execution, `smp09.exe.cov` file is generated.

Figure 7-1. Code Coverage Test

laysim-gr712rc-cli in Linux

```
[layright@CentOS RTEMS5.1-SMP]$ laysim-gr712rc-cli -r -core0 smp09.elf -cov
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - smp09.elf
   - .text, addr: 0x40000000, size 118192 bytes
   - .rtemsroset, addr: 0x4001CDB0, size 144 bytes
   - .rtemsstack, addr: 0x4001CE40, size 32768 bytes (not loaded, just info.)
   - .data, addr: 0x40024E40, size 1328 bytes
   - .bss, addr: 0x40025380, size 17952 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000


*** BEGIN OF TEST SMP 9 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
 CPU 1 start task TA1
 kill 10 clock ticks
--------------------------------------------------------------------------------
                          CPU USAGE BY THREAD
------------+---------------------------------------+--------------+---------
 ID         | NAME                                  | SECONDS      | PERCENT
------------+---------------------------------------+--------------+---------
 0x09010001 | IDLE                                  |     1.171035 |  91.028
 0x09010002 | IDLE                                  |     1.000131 |  76.363
 0x0a010001 | UI1                                   |     0.332825 |  24.968
 0x0a010002 | TA1                                   |     0.185180 |  13.654
------------+---------------------------------------+--------------+---------
 TIME SINCE LAST CPU USAGE RESET IN SECONDS:                       1.356217
--------------------------------------------------------------------------------

*** END OF TEST SMP 9 ***


CORE#1 halts because of Error Mode [64222487cycles/0.570179sec]

 Performance statistics for CORE#0
   Cycles       : 64222527
   Instructions : 215588
   Overall CPI  : 5.09
   CORE#0 performance (100MHz) : 19.65 MOPS (19.65 MIPS, 0.00 MFLOPS)
   Simulated time              : 642.23 ms
   Processor utilization       : 1.71 %

 Performance statistics for CORE#1
   Cycles       : 64222487
   Instructions : 32157740
   Overall CPI  : 2.00
   CORE#1 performance (100MHz) : 50.07 MOPS (50.07 MIPS, 0.00 MFLOPS)
   Simulated time              : 642.22 ms
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 112.64 %
   laysim-gr712rc performance  : 56.78 MIPS
   Wall clock                  : 0.57 s

[layright@CentOS RTEMS5.1-SMP]$ ls -als smp09.elf.cov
56 -rw-rw-r--. 1 layright layright 53552 Sep 26 19:30 smp09.elf.cov
[layright@CentOS RTEMS5.1-SMP]$
```

laysim-gr712rc-cli in Windows

```
PS C:\laysim-gr712rc\RTEMS5.1-SMP> laysim-gr712rc-cli -r -core0 smp09.elf -cov
laysim for GR712RC Dual Core Processor CLI v3.1 by layright

CORE#0 RAM file is loaded - smp09.elf
   - .text, addr: 0x40000000, size 118192 bytes
   - .rtemsroset, addr: 0x4001CDB0, size 144 bytes
   - .rtemsstack, addr: 0x4001CE40, size 32768 bytes (not loaded, just info.)
   - .data, addr: 0x40024E40, size 1328 bytes
   - .bss, addr: 0x40025380, size 17952 bytes (not loaded, just info.)
   - Detect RTEMS v5.1(SMP) with thread information
Only GR712RC CORE#0 RAM file is loaded, starts from RAM
Set CORE#0 %sp/%fp : 0x407FFFF0/0x40800000


*** BEGIN OF TEST SMP 9 ***
*** TEST VERSION: 5.1.0
*** TEST STATE: EXPECTED_PASS
*** TEST BUILD: RTEMS_NETWORKING RTEMS_POSIX_API RTEMS_SMP
*** TEST TOOLS: 7.5.0 20191114 (RTEMS 5, RSB 5.1, Newlib 7947581)
 CPU 1 start task TA1
 kill 10 clock ticks
--------------------------------------------------------------------------------
                          CPU USAGE BY THREAD
------------+---------------------------------------+--------------+---------
 ID         | NAME                                  | SECONDS      | PERCENT
------------+---------------------------------------+--------------+---------
 0x09010001 | IDLE                                  |     1.171035 |  91.028
 0x09010002 | IDLE                                  |     1.000131 |  76.363
 0x0a010001 | UI1                                   |     0.332825 |  24.968
 0x0a010002 | TA1                                   |     0.185180 |  13.654
------------+---------------------------------------+--------------+---------
 TIME SINCE LAST CPU USAGE RESET IN SECONDS:                       1.356217
--------------------------------------------------------------------------------

*** END OF TEST SMP 9 ***


CORE#1 halts because of Error Mode [64222487cycles/0.670941sec]

 Performance statistics for CORE#0
   Cycles       : 64222527
   Instructions : 215588
   Overall CPI  : 5.09
   CORE#0 performance (100MHz) : 19.65 MOPS (19.65 MIPS, 0.00 MFLOPS)
   Simulated time              : 642.23 ms
   Processor utilization       : 1.71 %

 Performance statistics for CORE#1
   Cycles       : 64222487
   Instructions : 32157740
   Overall CPI  : 2.00
   CORE#1 performance (100MHz) : 50.07 MOPS (50.07 MIPS, 0.00 MFLOPS)
   Simulated time              : 642.22 ms
   Processor utilization       : 100.00 %

 Performance of laysim-gr712rc
   Real-time performance       : 95.72 %
   laysim-gr712rc performance  : 48.25 MIPS
   Wall clock                  : 0.67 s

PS C:\laysim-gr712rc\RTEMS5.1-SMP> ls smp09.elf.cov


    Directory: C:\laysim-gr712rc\RTEMS5.1-SMP


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----        9/26/2025   7:32 PM          54246 smp09.elf.cov


PS C:\laysim-gr712rc\RTEMS5.1-SMP>
```

```
40000000 : 7 3 3 3 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 1 19 19 0 1 1 1 1 6 2 2 2
40000080 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000100 : 6 6 6 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000180 : 7 3 3 3 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 7 3 3 3 6 2 2 2
40000200 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000280 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000300 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000380 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000400 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000480 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000500 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000580 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000600 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000680 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000780 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
40000800 : 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000880 : 6 2 2 2 19 19 19 19 19 19 19 19 9 1 9 19 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000900 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000980 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000a00 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
```

```
40000000 : 7 3 3 3 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 1 19 19 0 1 1 1 1 6 2 2 2
40000080 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000100 : 6 6 6 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000180 : 7 3 3 3 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 7 3 3 3 6 2 2 2
40000200 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000280 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000300 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000380 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000400 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000480 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000500 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000580 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000600 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000680 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000780 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
40000800 : 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000880 : 6 2 2 2 19 19 19 19 19 19 19 19 9 1 9 19 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000900 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000980 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000a00 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
```

```
40000a80 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000b00 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000b80 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
...
...
4002ec80 : 2 2 2 2 2 2 0 6 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2
4002ed00 : 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2
4002ed80 : 2 2 0 6 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2 2 2 2 2 2
4002ee00 : 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6
4002ee80 : 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2 6 2 2 2 2 2 2 2
4002ef00 : 2 2 2 2 2 2 0 6 6 6 6 2 6 2 6 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2
4002ef80 : 6 2 2 2 0 6 6 6 2 2 6 2 6 2 0 6 6 6 2 2 6 2 6 2 0 6 6 6 2 2 6 6
4002f000 : 6 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2
4002f080 : 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6
4002f100 : 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6
4002f180 : 2 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2
4002f200 : 6 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2 6 2 6 2 2 2 2 2 2 2 2 2 2
4002f280 : 2 2 0 6 6 6 2 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2
4002f300 : 0 6 6 6 2 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6
4002f380 : 6 6 2 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 2 0 6 6
4002f400 : 2 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 2 0 6 6 6 2 2
4002f480 : 2 2 2 2 2 0 6 6 6 2 2 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2 6 2
4002f500 : 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2
4002f580 : 0 6 6 6 2 2 6 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2 6 2 2 2 2 2 2
4002f600 : 2 2 2 2 2 2 2 2 2 0 6 6 6 6 6 2 2 2 6 2 6 6 2 2 2 2 2 2 2 6 6 6 6
4002f680 : 6 2 2 2 0 6 6 6 2 6 6 6 6 6 6 6 6 6 6 6 2 6 2 6 6 6 0 0 2 2 2 2
407fff80 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 6 0 0 0 0
```

```
40000a80 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000b00 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
40000b80 : 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2 6 2 2 2
...
...
4002ec80 : 2 2 2 2 2 2 0 6 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2
4002ed00 : 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2
4002ed80 : 2 2 0 6 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2 2 2 2 2 2
4002ee00 : 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6
4002ee80 : 6 6 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 6 2 6 2 2 2 2 2 2 2
4002ef00 : 2 2 2 2 2 2 0 6 6 6 6 2 6 2 6 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2
4002ef80 : 6 2 2 2 0 6 6 6 2 2 6 2 6 2 0 6 6 6 2 2 6 2 6 2 0 6 6 6 2 2 6 6
4002f000 : 6 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2
4002f080 : 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6
4002f100 : 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6
4002f180 : 2 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2
4002f200 : 6 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2 6 2 6 2 2 2 2 2 2 2 2 2 2
4002f280 : 2 2 0 6 6 6 2 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2
4002f300 : 0 6 6 6 2 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6
4002f380 : 6 6 2 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 2 0 6 6
4002f400 : 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2 2 2 2 2 0 6 6 6 2 2
4002f480 : 2 2 2 2 0 6 6 6 2 2 6 2 2 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2 6 2
4002f500 : 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2 6 2 2 2 2 2 2 2 2 2 2 2 2 2
4002f580 : 0 6 6 6 2 2 6 2 2 2 2 2 2 2 2 2 2 2 2 0 6 6 6 2 2 6 2 2 2 2 2
4002f600 : 2 2 2 2 2 2 2 2 0 6 6 6 6 6 2 2 2 6 2 6 6 2 2 2 2 2 2 2 6 6 6 6
4002f680 : 6 2 2 2 0 6 6 6 2 6 6 6 6 6 6 6 6 6 6 6 2 6 2 6 6 6 0 0 2 2 2 2
407fff80 : 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 6 0 0 0 0
```

8. Emulation Characteristics of laysim-GR712RC

This chapter describes which features of the GR712RC are supported and not supported in `laysim-GR712RC`.

8.1 LEON3 Support

`laysim-GR712RC` supports most of the features of the LEON3 in GR712RC, which can be summarized as follows.

- Integer Unit (IU) : `laysim-GR712RC` supports the LEON3 integer unit with hardware multiply and divide instruction, and eight register windows.
  - ► Supported
    - Eight register windows
    - Hardware multiplier and Radix-2 divider
    - Processor configuration register (ASR17) as read-only register
    - Address space identifiers (ASI)
    - Power-down mode : `laysim-GR712RC` supports the power-down mode of GR712RC, and `laysim-GR712RC` can increase the simulation speed because it skips time until the next event on power-down mode. Also `laysim-GR712RC` supports the idle-loop optimization, where it acts like the power-down mode.
    - Configuration registers : `laysim-GR712RC` supports ASR17. ASR16 is always zero, and ASR24-27 are not supported because of the support breakpoints/watchpoint in emulator level.
    - Single-vector trapping for reduced code size
  - ► Unsupported
    - Register file data protection : ASR16 is always zero
    - Hardware breakpoints : `laysim-GR712RC` supports breakpoints and watchpoints at the emulator level.
    - Instruction trace buffer : `laysim-gr712rc` supports history operations at the emulator level.

- Cache sub-system : Only `laysim-gr712rc` supports L1 cache for instruction and data of the LEON3 with L1 cache hit/miss cycles and write buffer cycles. It also supports the data cache snooping, but it does not support the diagnostic access for the cache tag and data. `laysim-gr712rc-dbt` does not model the L1 cache of GR712RC for speed-up.
  - ► Supported (Only `laysim-gr712rc`)
    - Enabling and disabling cache
    - Cache freeze and flushing
    - Data cache snooping
    - Operating with MMU
  - ► Unsupported (Only `laysim-gr712rc`)
    - Diagnostic access : The cache tag and data can be checked through the Cache Window of `laysim-gr712rc`.

- Memory Management Unit (MMU) : `laysim-GR712RC` supports the MMU of GR712RC including a three-level hardware table-walk, 16 TLBs, and MMU registers.

- Floating Point Unit (FPU) : `laysim-GR712RC` supports the GRFPU with the simple FPU cycle model. It supports the handling

denormalized numbers of the GRFPU, and supports only a single FPU trap queue.

8.2 FTMCTRL Support

FTMCTRL does not provide actual functionality and only provides registers and P&P information in `laysim-GR712RC`. When the SRAM/MRAM size is set with `-ramsz`/`-romsz`, and it is reflected in FTMCTRL. The memory mapped I/O is supported through a loadable module. More information can be found in `[laysim-GR712RC-005] laysim-GR712RC User Loadable Modules - User's Manual`.

8.3 FTAHBRAM Support

Any DMA master such as SpW2, 1553BRM can use FTAHBRAM as the shared memory, but it cannot be used for code area in `laysim-GR712RC`.

8.4 AHBSTAT Support

`laysim-GR712RC` supports the AHBSTAT with interrupts.

8.5 IRQMP Support

`laysim-GR712RC` supports the IRQMP of GR712RC with interrupt prioritization, extended interrupts, processor status monitoring, and interrupt broadcasting.

8.6 GPTIMER Support

`laysim-GR712RC` supports a GPTIMER of GR712RC without watchdog functionality.

8.7 GRTIMER Support

`laysim-GR712RC` supports a GRTIMER of GR712RC with time latch capability.

8.8 GRGPREG Support

`laysim-GR712RC` does not support GRGPREG and only provides dummy registers.

8.9 GRGPIO Support

`laysim-GR712RC` supports GRGPIOs including I/O port, interrupt generation and interrupt map. More information can be found in `[laysim-GR712RC-005] laysim-GR712RC User Loadable Modules - User's Manual`.

8.10 APBUART Support

`laysim-GR712RC` supports APBUARTs with Tx/Rx FIFO and loadable UART Module. But it doesn't support the external clock, loop back, parity, and break signal. More information can be found in `[laysim-GR712RC-005] laysim-GR712RC User Loadable Modules - User's Manual`.

8.11 SpaceWire Support

`laysim-GR712RC` supports two GRSpW2 with RMAP and four GRSpW2 without RMAP. Also it supports time-codes and loadable SpW Nodes. More information can be found in `[laysim-GR712RC-005] laysim-GR712RC User Loadable Modules - User's Manual`.

8.12 GRETH_GBIT Support

laysim-GR712RC fully supports GRETH with DMA. In addition, it provides UDP connectivity by default to facilitate integration with Tornado/Workbench. It can configure GRETH_GBIT for AIR and NASA cFS using '-air' or '-cfs' start-up option.

8.13 CANOC Support

laysim-GR712RC supports OCCANs with BasicCAN and PeliCAN including CAN protocol, message transmission, filtering and reception, and interrupt generation. More information can be found in [laysim-GR712RC-005] laysim-GR712RC User Loadable Modules - User's Manual.

8.14 CANMUX Support

laysim-GR712RC does not support CANMUX and only provides dummy registers.

8.15 1553BRM Support

The GR712RC provides Actel Core 1553BRM with B1553BRM (0x01:0x072) for MIL-STD-1553B interface. laysim-GR712RC supports BC mode without supporting RT-to-RT transfer and RT using loadable 1553B modules. More information can be found in [laysim-GR712RC-005] laysim-GR712RC User Loadable Modules - User's Manual.

8.16 I2C Support

laysim-GR712RC supports the I2C master with loadable module. More information can be found in [laysim-GR712RC-005] laysim-GR712RC User Loadable Modules - User's Manual.

8.17 SPI Support

laysim-GR712RC supports the SPI controller with loadable module. More information can be found in [laysim-GR712RC-005] laysim-GR712RC User Loadable Modules - User's Manual.

8.18 SLINK Support

laysim-GR712RC does not support SLINK and only provides dummy registers.

8.19 ASCS Support

laysim-GR712RC does not support ASCS and only provides dummy registers.

8.20 GRTC Support

laysim-GR712RC does not support GRTC and only provides dummy registers.

8.21 GRTM Support

laysim-GR712RC does not support GRTM and only provides dummy registers.

8.22 GRCLKGATE Support

In laysim-GR712RC, the GRCLKGATE is checked only when the corresponding core is enabled. If the GRCLKGATE is disabled, an error

is logged. During a read operation, the register can be accessed normally, but a write operation is disabled. When a RAM executable is loaded, the GRCLKGATE of all cores is automatically enabled.