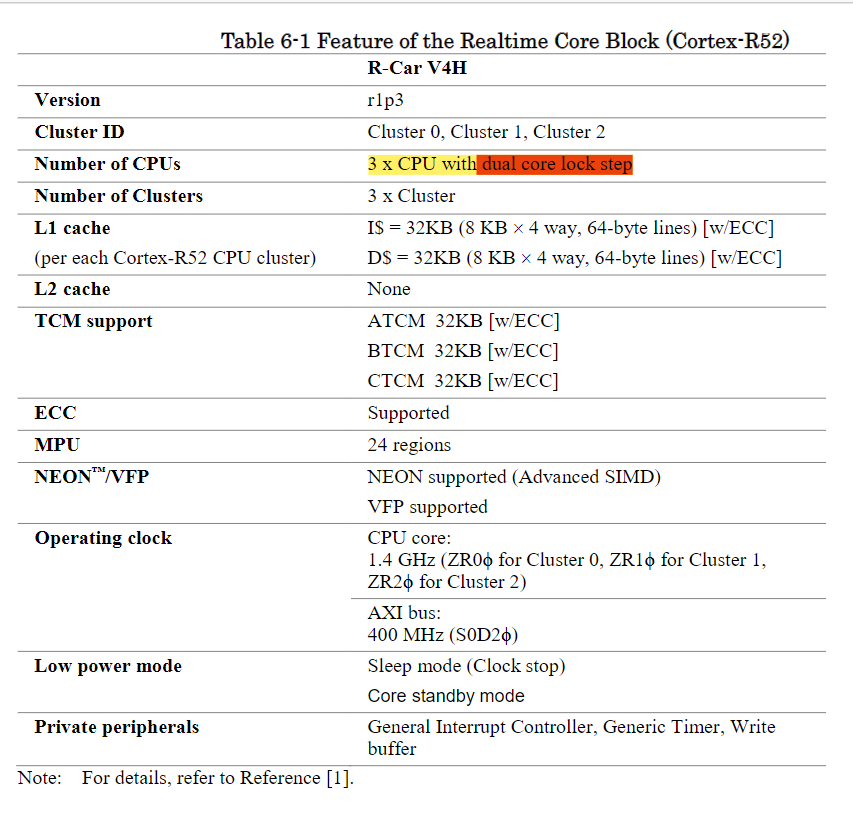
# **1 Overview**

The Realtime core is a core block equipped with Arm® Cortex®-R52

## **Features**

Feature of the Realtime Core Block (Cortex-R52)



## **Block Diagram**

Diagram

Description automatically generated

## **Register Configuration**

Graphical user interface, text, application

Description automatically generated

Table

Description automatically generated

* 1. **Connected Module**

Table

Description automatically generated with medium confidence

# **Register Description**

* 1. **Write Buffer Control Register**

WBCTLR is the register to enable the write buffer function

## **Write Buffer Interrupt Mask Register**

WBIMSKR is the register to enable interrupt when a bus error is detected

## **Write Buffer Interrupt Mask Status Register**

WBIMSKSTSR is the register to hold the error status after being masked by WBIMSKR

## **Write Buffer Error Status Register**

WBERRSTSR is the register to hold the error status before being masked by WBIMSKR

## **Write Buffer Interrupt Clear Register**

WBICLRR is the register to clear error status.

* 1. **Write Buffer SLVERR Address Register**

WBSERRADDR is the register to hold the address at which SLVERR is detected. The value of this register is valid only when SLVERRSTS bit in WBERRSTSR register is set to 1.

## **Write Buffer DECERR Address Register**

WBDERRADDR is the register to hold the address at which DECERR is detected. The value of this register is valid only when DECERRSTS bit in WBERRSTSR register is set to 1.

## **Write Buffer Sync Operation Register**

WBSYNCR is the register to execute memory barrier operation accompanied with the data eviction from write buffer. This register is automatically cleared by hardware when all the responses from the target module are returned.

## **Power Control Register**

WBPWRCTLR is the register to control the dynamic module stop function.

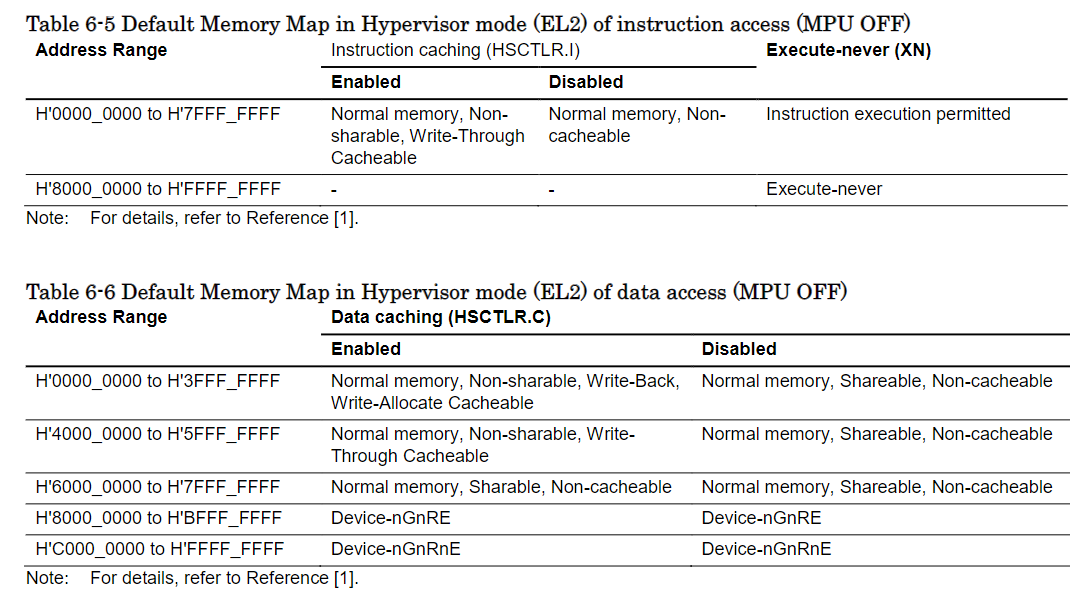
# **Operation**

* 1. **Boot Address Setting**

The boot address of Realtime Core can be controlled by Boot Address Register (BAR) in Advanced Power Management Unit (APMU) module.

* 1. **MPU (Memory Protection Unit) setting**

The default memory map of Realtime Core



Graphical user interface, application

Description automatically generated with medium confidence

Table

Description automatically generated

* 1. **Power-Up/Down Mechanism**

The Realtime Core supports the following three power modes.  
• Core Standby mode  
• Sleep mode (clock stop)  
• Normal operation mode

* 1. **Write Buffer**

The Write Buffer is a buffer equipped between Cortex-R52 core and AXI master port, the write requests which the target address are successive can be merged by using the write buffer. Write buffer has three merge slots. Each merge slot can store up to 128-byte data.

Once the write data is stored to the write buffer, the write response is returned to the bus master interface of

Realtime Core before receiving the response from the target modules. When the corresponding response from the target modules are returned with the bus error attribute (DECERR/SLVERR), the interrupt related to this error can be asserted and notified to ECM.

To enable the write buffer, Realtime Core sets 1’b1 to WBCTLR register. For data eviction operation, there are two cases. In the first case, automatic eviction occurs when buffer full or timeout. In the second case, Realtime Core sets 1’b1 to WBSYNCR register to execute memory barrier operation accompanied with the data eviction from the write buffer. In order to complete the write transactions when the write buffer is used, data eviction operation from the write buffer is needed

* 1. **Private Peripheral Modules**

The private peripheral modules in Realtime Core are operated at ZR0φ, ZR1φ and ZR2φ

Base address of private peripheral modules is defined by hardware. In R-Car V4H products, the base address is “H’F000 0000”.

* 1. **Configuration signals setting**

Table

Description automatically generated with medium confidence

**4 Usage Notes**

**4.1 Interrupt handling for Realtime Core**

Cortex-R52 does not support NMI (Non-Maskable Interrupt). EXTPPI[0] is used to connect NMI pin from INTC-EX.

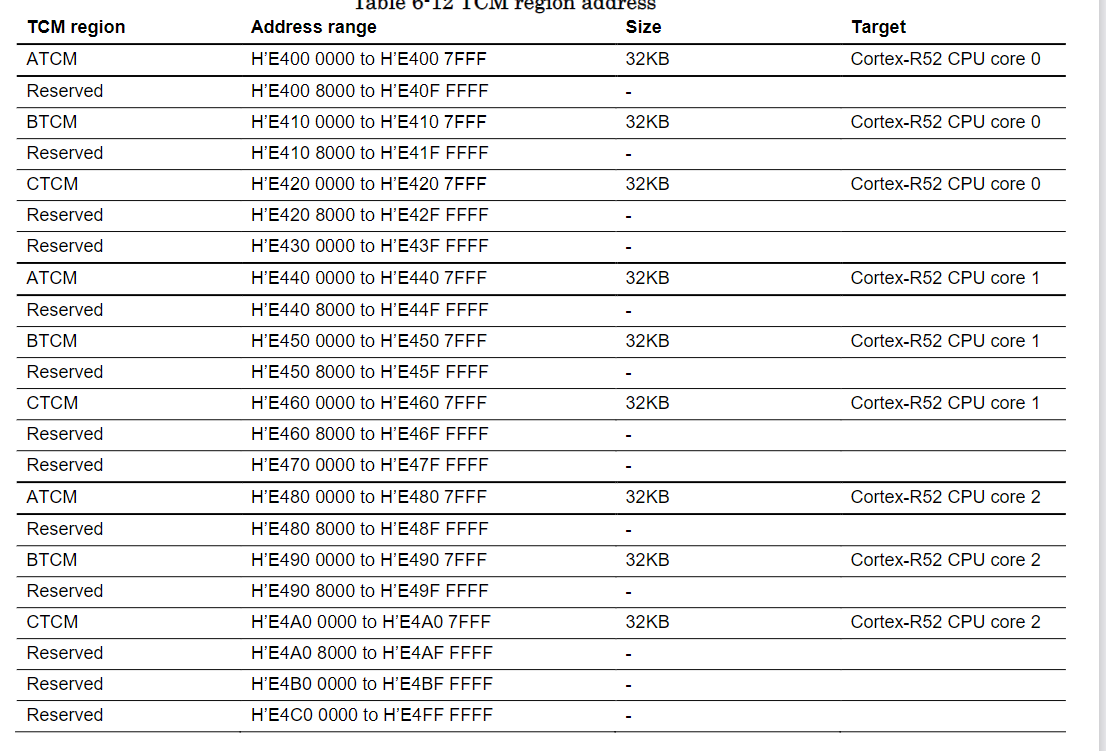
**4.2 Accessing to TCM regions**

**4.2.1 Accessing to TCM regions by CPU**

Realtime Core can access to its TCMs directly. Software has to set the base address, size of each TCM and enable it by writing to IMP\_ATCMREGIONR, IMP\_BTCMREGIONR, IMP\_CTCMREGIONR. Base address must be size-aligned.

**4.2.2 Accessing to TCM regions via AXIS interface**

The TCM address range (16MB region) defined in system address map is from H’E400 0000 to H’E4FF FFFF. Those regions are accessed via AXIS port. Accessing to reserved areas will cause DECERR.



Access control checks can be enabled for AXIS transactions by programming  
IMP\_SLAVEPCTLR.TCMACCLVL. Access control allows either all transactions or only privileged transactions to access the TCM. If a transaction is not permitted, the AXIS generates a SLVERR response.  
Privileged of transactions that access TCM can be overwritten by Cortex-R52 Secure Auxiliary Configuration

**4.3 Watchdog Time**

Cortex-R52 does not support watchdog timer. Consider choosing watchdog timers outside of Cortex-R52

**4.4 Exclusive operation usage restriction**

Exclusive operation using LDREX and STREX instruction pair is supported only in SDRAM area (H'4000 0000 to H'BFFF FFFF) and System RAM area. Exclusive operation to other areas is not supported

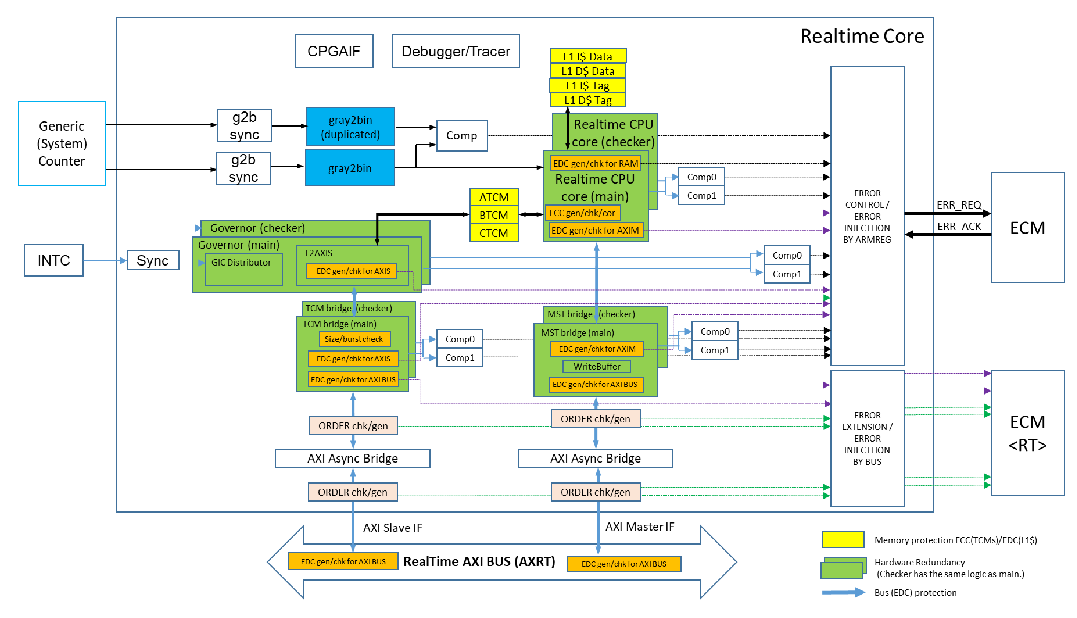
**4.5 AxPROT[1] overwrite function on AXI Master interface**

Cortex-R52 always issues secure transactions (AxPROT[1] = 0) to AXI Master interface. Realtime Core has feature to change from secure transactions to Non-secure transactions (AxPROT[1] = 1) by writing 0 to register (address 0xFFC43018). This register can be changed from 1 (default value) to 0 only. Before changing this register from 1 to 0, it’s mandatory to flush all data in Cortex-R52 Store Buffer and in Write Buffer to external memory. Software can use “DSB” instruction to flush all data in Cortex-R52 Store Buffer, then use Write Buffer Sync Operation Register to flush all data in Write Buffer to external memory

# **5 Safety Mechanisms**

## **5.1 Overview**

Realtime Core system is equipped with various Safety Mechanism.



## **5.2 Error Correction Code (ECC)**

ECC protection is applied to the following RAMs.  
• L1 D-cache Tag RAM  
• L1 D-cache Data RAM  
• L1 I-cache Tag RAM  
• L1 I-cache Data RAM  
• ATCM  
• BTCM

## **5.3 Bus Protection**

Signal Integrity Protection is implemented to protect AXI Interface between Realtime Core CPU & AXI  
bus bridges.  
• Bus timeout detection is implemented in AXI-bus master Interface to detect transactions that fail to  
complete within a programmable time limit.  
• EDC protection is implemented to protect AXI Interface between Realtime Core and RT (Realtime)  
domain AXI (AXRT).  
• ORDER check is implemented to detect faults of FIFO read/write pointers in AXI Asynchronous  
bridges.  
• SIZE-BUSRT check is implemented in AXI Slave interface (AXIS) to detect illegal AxSIZE/AxBURST  
transactions.

## **5.4 Hardware Redundancy (DCLS)**

The dual core lock step (DCLS) architecture is implemented in Realtime Core.

The function is realized by running duplicated CPU core, write buffer. The checker core has the same input of main core, and the checker write buffer has the same input of main write buffer.  
When the output of main core and checker core, and/or the output of main write buffer and checker write buffer are different, an error is detected. The Realtime Core sends a notification of the error to the ECM module. To enable the DCLS operation, writing 1 to bit 0 of CR52CMPEN register is required

## **5.5 Generic Counter Interface Protection**

Hardware Redundancy is applied to detect if a fault of Generic Counter Interface between Generic Counter and Realtime Core.

## **5.6 Address Feedback**

Cortex-R52 has method to detect the faults in the memory address decoder that are built into ECC scheme

## **5.7 Other Error notification**

Realtime Core supports to notify several types of Errors/Events to ECM. Software can use these  
Errors/Events notification as needed