User's Manual

COMA
(R-Switch-3 GateWay Common Agent)

Document number : 0.50
Date of issue : 2025/02/18

- This document and contents shall be strictly managed for all document users and used for specific/limited purpose only, issued under NDA between Renesas Electronics and the company of this document holders
- Under development
- Preliminary document
  Specification in this document are tentative and subject to change
- Renesas Electronics Confidential - issued under NDA

ルネサス エレクトロニクス
www.renesas.com

Notice

1.Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3.Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of information of product data, diagrams, tables, programs, algorithms or application circuit example etc. described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4.You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5.Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The recommended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
“Standard”: Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots etc.
“High Quality”: Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anticrime systems; and safety equipment etc.
Renesas Electronics products are neither intended nor authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems, surgical implantations etc.), or may cause serious property damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6.You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. Please note that Renesas Electronics shall not be responsible for any damage if products are not used as per the various conditions stated in this manual, due to resale by customer etc.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.

Note 1. “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.
Note 2.“Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(2012.4)

General precautions for handling of product

The following notes are applicable to entire CBIC with CPU core. For detailed usage notes, refer to the relevant sections of the manual. If the description under General precautions and in the body of the manual differs from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.

- The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flow internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.

2. Processing at Power-on

The state of the product is undefined at the moment when power is supplied.

- The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.

3. Prohibition of Access to Reserved Addresses Access to reserved addresses is prohibited.

- The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

4. Regarding Clock

After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized. When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.

Table of Contents

1. Overview

Common agent is an agent of R-Switch system which aims at gathering common functionalities for other agents.

It handles the APB to SFR bus conversion, the buffer pointer release for rejected frames and the pointer handling for Local RAM. SW involvement required for common agent configuration.

1.1 Features

Common Agent Features are described in Table 1-1.

Table 1-1 Common Agent Feature List

Function		Details
Overall function	Pointer management	・Manage the Local RAM pointers and their distribution
	APB to SFR	・Convert APB protocol in SFR bus ・Distribute SFR access among other switch IPs
	Descriptor reject	・Reject the descriptors unused by agents for transmission
Data provision	Fabric error bus	・Pointer received from fabric through error bus
	Descriptor bus	・Descriptor received from forwarding engine [FWD] for rejection
Control function	RAM access	・Function that writes and reads to RAM

1.2 Common agent block diagram

Fig 1.1 shows common agent block diagram.

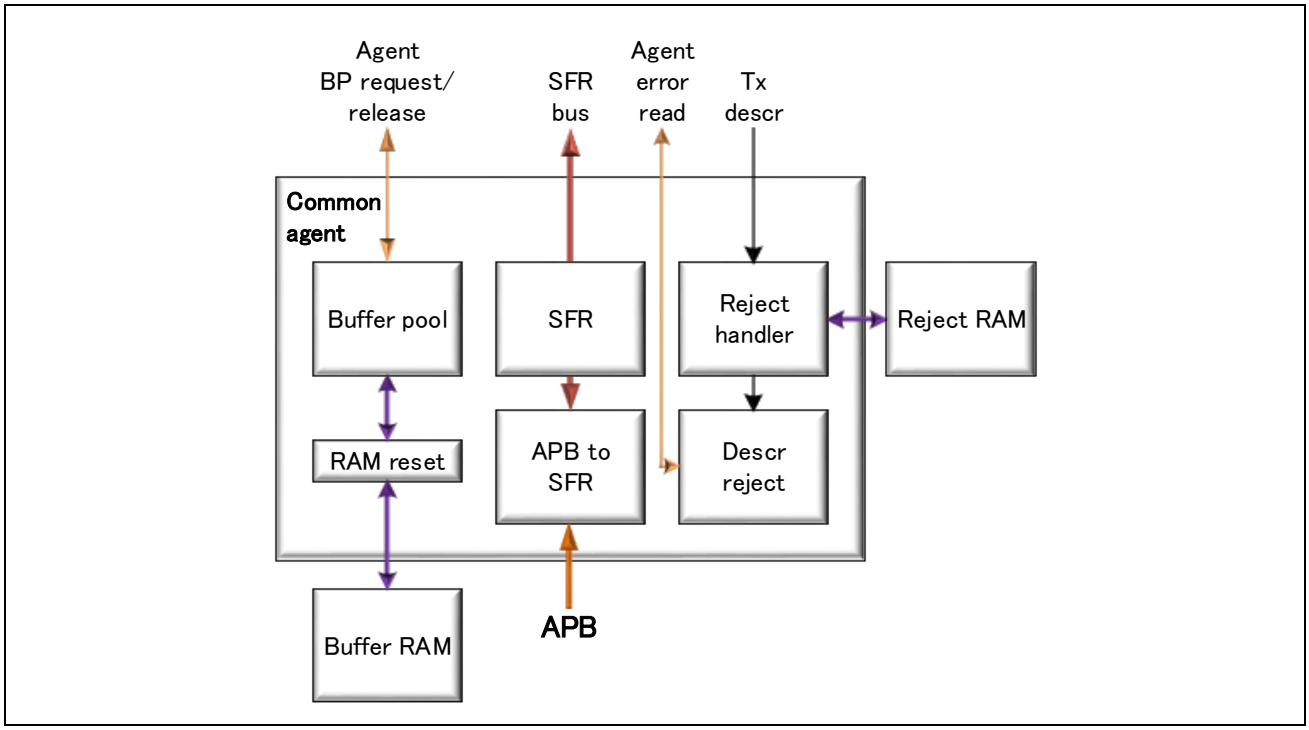


Fig 1.1: Common agent Block diagram

Table 1-2 Common Agent Functional Blocks

Block name	Function
APB to SFR	Convert APB bus to SFR bus for every IP in R-Switch system
SFR	Common agent SFR
Reject handler	Receive descriptors to be rejected from Forwarding Engine and save them in reject RAM
Descr reject	Reject the descriptors
Buffer pool	Handles the free buffer pointers
RAM reset	Handle RAM reset

2. Parameter list

Common Agent global parameter list is shown in Table 2-1

Common Agent local parameter list is shown in Table 2-2

Table 2-1 Global parameter list

Parameter Name	**RW3 Values**	**Explanation**
SFRs		
PADDR_NBR_FWD	**110592/4**	Number of addresses used by Forwarding Engine SFRs[FWD]
PADDR_NBR_FAB	4096/4	Number of addresses used by Fabric SFRs[FAB]
PADDR_NBR_COMA	4096/4	Number of addresses used by Common Agent SFRs
PADDR_NBR_GWCA	8192/4	Number of addresses used by GWCA SFRs[GWCA]
PADDR_NBR_TSNA	8192/4	Number of addresses used by TSN Agent SFRs[TSN]
Port Number		
PORT_TSNA_N	**13**	TSN Agent Number [TSNA]
PORT_GWCA_N	**2**	CPU Agent Number [GWCA]
Local RAM		
LCL_RAM_SZ	1024	Local RAM size in Kbytes
LCL_RAM_BSZ	128	Local RAM block size (A pointer will always link to a LOCAL_RAM_BSZ byte block size in the local RAM)
LCL_RAM_ECCR_N	8	Local RAM ECC error recovery number. Number of ECC error losing pointer allowed in the Buffer pool.
Frame		
FRM_TPL_W	16	Frame TPL (Total payload length) Width
Counter		
COUNT_MED_W	32	Medium size Counter width
Pause frame		
PAS_LVL_N	2	Pause level number

Table 2-2 Local parameter list

Parameter Name	RSW3 Values	Explanation
SFRs		
PADDR_NBR_FWD_W	15	Number of addresses used by Forwarding Engine SFRs bus width[FWD]
PADDR_NBR_FAB_W	10	Number of addresses used by Fabric SFRs bus width[FAB]
PADDR_NBR_COMA_W	10	Number of addresses used by Common Agent SFRs bus width
PADDR_NBR_GWCA_W	11	Number of addresses used by GWCA SFRs bus width[GWCA]
PADDR_NBR_TSNA_W	11	Number of addresses used by TSN Agent SFRs bus width[TSN]
PADDR_AW	16	APB address bus width (the two APB unused bits are not taken in account)
Port Number		
PORT_N	**16**	Port number on the switch
PORT_W	4	Port number on the switch bus width
PORT_W1	5	Port number on the switch +1 bus width
Local RAM		
LCL_PTR_N	8192	Pointer number to address local RAM
LCL_PTR_W	13	Pointer width
LCL_PTR_W1	14	Pointer+1 width
LCL_RAM_ECCR_W	3	Local RAM ECC error recovery bus width
LCL_RAM_ECCR_W1	4	Local RAM ECC error recovery +1 bus width
Frame		
FRM_MTN_W	5	Port number on the switch bus width
DEF_FRAME_SIZE	2048	Default frame size (bytes)
Buffer RAM		
BPR_RAM_DW	18	Buffer pool RAM data bus width
Reject RAM		
RJT_RAM_DW	43	Reject RAM bus width

3. Register

3.1 Register attributes

The register attribute defines what kind of access a register supports. Per one register, there are always two attributes, a register access attribute which define what kind of accesses a register supports and, a register security attribute which define what accesses can perform the unsecure APB [APB] in the register access attribute depending on the security setting in security registers.

“Representation of register access attributes “ describes register access attributes and “Representation of register security attributes” describes register security attributes. Attributes are given to a register field in Register detailed explanation section by specifying the attribute symbols in the R/W-P column.

Table 3-1: Register access attributes

Symbol	Meaning	Impact on accesses	
		Write access	Read access
RW	Read write	Write value is written	Written value is read
R!=W	Read different than write	Write access happens	Read value differs from written value
R	Read only	Write value is ignored	Read access happens
R0	Only Read 0	Write value is ignored	Always read '0'
R1	Only Read 1	Write value is ignored	Always read '1'
R0W	Read 0 write	Write access happens	Always read as '0'
R1W	Read 1 write	Write access happens	Always read as '1'
RC	Read clear	Write value is ignored	Read access happens Read access clears the register

Table 3-2: Register security attributes (For R-Car products only)

Symbol	Meaning	Impact on accesses	
		Write access	Read access
U	Unprotected	Write access happens for unsecure APB	Read access happens for unsecure APB
P	Protected	A security register should be set to authorize write access by the unsecure APB.	A security register should be set to authorize read access by the unsecure APB
RU	Read-Unprotected	Write value ignored for unsecure APB	Read access happens for unsecure APB
RP	Read protected	Write value ignored for unsecure APB	A security register should be set to authorize read access by the unsecure APB
D	Duplicated	Write access happens for unsecure APB to a duplicated and independent register	Read access happens for unsecure APB to a duplicated and independent register
F	Forbidden	Write value ignored for unsecure APB	Read value ignored for unsecure APB
S	Switch	A security register should be set to authorize write access by the unsecure APB. A security register should be set to unauthorize write access by the secure APB.	A security register should be set to authorize read access by the unsecure APB

3.2 Register list

The Common agent register list is described in Table 3-3. CARO (Common agent Register Offset) indicates base address of address space allocated to Common agent by the system. All registers representations are done with the default values of the section 2. If the COMA is not use with default parameters, it should be taken in account by the user while reading the SFR representation.

Note:

- A register can have two addresses. The address preceded by “E:” correspond to an emulation address which allows to read a register without modifying its content.

Table 3-3: List of Common Agent registers

Offset/Address	Register name	Abbreviation
CARO + 0000H	R-Switch IP Version	RIPV
CARO + 0004H	R-Switch Reset Configuration	RRC
CARO + 0008H	R-Switch Clock Enable Configuration	RCEC
CARO + 000CH	R-Switch Clock Disable Configuration	RCDC
CARO + 0010H	R-Switch Software Synchronization Interrupt Status	RSSIS
CARO + 0014H	R-Switch Software Synchronization Interrupt Enable	RSSIE
CARO + 0018H	R-Switch Software Synchronization Interrupt Disable	RSSID
CARO + 0020H + 4*i	Common Agent Buffer Pool IPV Based Watermark Configuration i (i=0..7)	CABPIBWMCi
CARO + 0040H	Common Agent Buffer Pool Watermark Level Configuration	CABPWMLC
CARO + 0050H + 4*i	Common Agent Buffer Pointer Pause Frame Level Configuration i (i = 0..PAS_LVL_N -1)	CABPPFLCi
CARO + 0060H + 4*i	Common Agent Buffer Pool Watermark Level Configuration i (i = 0..PORT_N-1)	CABPPWMLCi
CARO + 00A0H + 4*(i*PAS_LVL_N + j)	Common Agent Buffer Pointer per Port Pause Frame Level Configuration i j (i = 0..PORT_N -1) (j = 0..PAS_LVL_N-1)	CABPPPFLCij
CARO + 0120H + 4*i	Common Agent Buffer Pointer Utilization Level Configuration i (i = 0..PORT_N-1)	CABPULCi
CARO + 0160H	Common Agent Buffer Pool Initialization Register Monitoring	CABPIRM
CARO + 0164H	Common Agent Buffer Pool Pointer Count Monitoring	CABPPCM
CARO + 0168H E: CARO + 016CH	Common Agent Buffer Pool Pointer Least Count Monitoring	CABPLCM
CARO + 0180H + 4*i	Common Agent Buffer Pointer Count per Port Monitoring i (i = 0..PORT_N-1)	CABPCPMi
CARO + 0200H + 4*i E: CARO + 0280H + 4*i	Common Agent Buffer Pointer Maximum Count per Port Monitoring i (i = 0..PORT_N-1)	CABPMCPMi
CARO + 0300H	Common Agent Rejected Descriptor Number Monitoring	CARDNM
CARO + 0304H E: CARO + 0308H	Common Agent Rejected Descriptor Maximum Number Monitoring	CARDMNM
CARO + 0310H E: CARO + 0314H	Common Agent Rejected Descriptor Counter	CARDCN
CARO + 0400H	Common Agent Error Interrupt Status 0	CAEIS0
CARO + 0404H	Common Agent Error Interrupt Enable 0	CAEIE0
CARO + 0408H	Common Agent Error Interrupt Disable 0	CAEID0
CARO + 0410H	Common Agent Error Interrupt Status 1	CAEIS1
CARO + 0414H	Common Agent Error Interrupt Enable 1	CAEIE1
CARO + 0418H	Common Agent Error Interrupt Disable 1	CAEID1
CARO + 0440H	Common Agent Monitoring Interrupt Status 0	CAMIS0
CARO + 0444H	Common Agent Monitoring Interrupt Enable 0	CAMIE0
CARO + 0448H	Common Agent Monitoring Interrupt Disable 0	CAMID0
CARO + 0450H	Common Agent Monitoring Interrupt Status 1	CAMIS1

Offset/Address	Register name	Abbreviation
CARO + 0454H	Common Agent Monitoring Interrupt Enable 1	CAMIE1
CARO + 0458H	Common Agent Monitoring Interrupt Disable 1	CAMID1
CARO + 0480H	Common Agent Security configuration register.	CASCR

Restrictions:

- If nothing specified, a register need an authorization from secure APB interface to be accessed by unsecure APB interface. This applies for Read and Write accesses. Any exception to this rule will be specified under “Security restrictions” or “Security un-restrictions” labels.

3.3 Register detailed explanation

This section describes SFR details.

3.3.1 R-switch registers

3.3.1.1 Version register

(1) RIPV

R-Switch IP Version.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV								CAIPV[3:0]				FBIPV[3:0]			
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EAIPV[3:0]				FWIPV[3:0]				GWIPV[3:0]				TIPV[3:0]			

Bits	Bit name	RW-P	Initial value	Function description
31:24	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
23:20	CAIPV	R-P	3H	Common Agent IP version number.
19:16	FBIPV	R-P	3H	Fabric IP version number [FAB].
15:12	EAIPV	R-P	3H	TSN Agent IP version number [TSNA].
11:8	FWIPV	R-P	3H	Forwarding Engine IP version number [FWD].
7:4	GWIPV	R-P	3H	GWCA IP version number [GWCA].
3:0	TIPV	R-P	3H	R-Switch Top IP version number [TOP].

3.3.1.2 Reset register

(1) RRC

R-Switch Reset Configuration.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV															RR

Bits	Bit name	RW-P	Initial value	Function description
31:1	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
0	RR	RW-F	0H	R-Switch software reset. Values: - 1'b0: R-Switch not reset. - 1'b1: R-Switch software (emergency) reset. Functions: - When set this bit reset all the R-Switch system except the APB interface and registers of [TOP]. It will be reset all registers. However, RAM and TCAM will not be reset. (i.e. Reinitialize the RAM and reconfigure the table) - The synchronous reset RCC.RR is power consuming, so it is internally only asserted 7 clocks. - The synchronous reset is also internally asserted 7 clocks after hardware reset release. Cautions: - The synchronous reset RCC.RR is power consuming. It should always be de-asserted soon after assertion to avoid power consumption.

3.3.1.3 Clock registers

(1) RCEC

R-Switch Clock Enable Configuration.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															RCE
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV / ACE[PORT_N-1:0]															

Bits	Bit name	RW-P	Initial value	Function description
31:17	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
16	RCE	R!=W-RU	0H	R-Switch Clock Enable Values: - 1'b0: R-switch clock disabled. - 1'b1: R-switch clock enabled. Functions: - Enables switch clocks. - When **RRC.RR** register is set, R-switch clock is internally enabled to allow software reset even if **RCC.RCE** is not set. - If RCE==0, MFWD APB read data always retains the state before the clock was stopped. Set conditions: - SW: Writing 1 to this bit will set it Clear conditions: - SW: Writing 1 to **RCDC.RCD** register will clear this bit.
15: PORT_N	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
PORT_N-1:0	ACE	R!=W-RU	0H	Agent Clock Enable Values: - Bit i set to 1'b0: Agent i clock disabled. - Bit i set to 1'b1: Agent i clock enabled. Functions: - Bit i enables agent i clock. For agent mapping details, refer to "Source port generation" of fabric specifications [FAB]. - When **RRC.RR** register is set, all agent i clock is internally enabled to allow software reset even if **RCC.ACE**[i] is not set. - When **RCC.RCE** register is not set, agent i clock is internally disabled even if **RCC.ACE**[i] is set. Set conditions: - SW: Writing 1 to bit i will set it Clear conditions: - SW: Writing 1 to **RCDC.ACD[i]** register will clear this bit.

[Notes]

R-Switch have three system clocks.

1) "Peripheral clock" which can be stopped only upper layer (MCU system).

2) "MFWD clock" means controlled by RCE.
3) "Agent clock" means controlled by ACEi.

[COMA] registers are controlled by “Peripheral clock”, which cannot be stopped, so read and write access will happen for [COMA] always. By the way it will be stopped by upper layer.

When peripheral clock is running and "Agent clock" is stopped (ACE[i] is 0).
- APB write access will NOT happen for [GWCA] and [TSNA]
- APB read access will happen for [GWCA] and [TSNA].

When peripheral clock is running and "MFWD clock" is stopped (RCE is 0).
- APB write access will NOT happen for [MFWD].
- APB read access will happen for [MFWD].
- “Agent clocks” are stopped.

(2) RCDC

R-Switch Clock Disable Configuration.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															RCD
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV / ACD[PORT_N-1:0]															

Bits	**Bit name**	**RW-P**	**Initial value**	**Function description**
31: PORT_N	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
16	RCD	R0W-F	0H	R-Switch Clock Disable Functions: - Writing 1 to this bit will clear **RCEC.RCE** register. Restrictions: - SW: This bit should only be written when all agents are disabled and when **CABPPCM.RPC** is equal to LCL_PTR_N.
15: PORT_N	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
PORT_N-1:0	ACD	R0W-F	0H	Agent Clock Disable Functions: - Writing 1 to this register bit i will clear bit i in **RCEC.ACE** register.

3.3.1.4 Secure-non-secure software synchronization registers.

(1) RSSIS

R-Switch Software Synchronization Interrupt Status.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSSSIS7	NSSSIS6	NSSSIS5	NSSSIS4	NSSSIS3	NSSSIS2	NSSSIS1	NSSSIS0	SNSSIS7	SNSSIS6	SNSSIS5	SNSSIS4	SNSSIS3	SNSSIS2	SNSSIS1	SNSSIS0

Bits	Bit name	RW-P	Initial value	Function description
31:16	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
i i = 8..15	NSSSISi	R!=W-U	0H	Non-Secure to Secure Synchronization Interrupt Status i. Set conditions: - SW: Writing 1 to this bit with APB unsecure interface will set it. Clear conditions: - SW: Writing 1 to this bit with APB secure interface will clear it.
i i = 0..7	SNSSISi	R!=W-U	0H	Secure to Non-Secure Synchronization Interrupt Status i. Set conditions: - SW: Writing 1 to this bit with APB secure interface will set it. Clear conditions: - SW: Writing 1 to this bit with APB unsecure interface will clear it.

(2) RSSIE

R-Switch Software Synchronization Interrupt Enable.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSSSIE7	NSSSIE6	NSSSIE5	NSSSIE4	NSSSIE3	NSSSIE2	NSSSIE1	NSSSIE0	SNSSIE7	SNSSIE6	SNSSIE5	SNSSIE4	SNSSIE3	SNSSIE2	SNSSIE1	SNSSIE0

Bits	Bit name	RW-P	Initial value	Function description
31:16	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
i i = 8..15	NSSSIEi	R!=W-U	0H	Non-Secure to Secure Synchronization Interrupt Enable i. Set conditions: - SW: Writing 1 to one of these bits with APB secure interface will set it. Clear conditions: - SW: Writing 1 to one of **RSSID.NSSSID** with APB secure interface will clear the corresponding bit in this register.
i i = 0..7	SNSSIEi	R!=W-U	0H	Secure to Non-Secure Synchronization Interrupt Enable i. Set conditions: - SW: Writing 1 to one of these bits with APB unsecure interface will set it. Clear conditions: - SW: Writing 1 to one of **RSSID.SNSSID** with APB unsecure interface will clear the corresponding bit in this register.

(3) RSSID

R-Switch Software Synchronization Interrupt Disable.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSSSID7	NSSSID6	NSSSID5	NSSSID4	NSSSID3	NSSSID2	NSSSID1	NSSSID0	SNSSID7	SNSSID6	SNSSID5	SNSSID4	SNSSID3	SNSSID2	SNSSID1	SNSSID0

Bits	Bit name	RW-P	Initial value	Function description
31:16	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
i i = 8..15	NSSSIDi	R0W-U	0H	Non-Secure to Secure Synchronization Interrupt Disable i. Set conditions: - SW: Writing 1 to one of these bits with the secure APB interface will clear the corresponding bit in **RSSIE.NSSSIE** register.
i i = 0..7	SNSSIDi	R0W-U	0H	Secure to Non-Secure Synchronization Interrupt Disable i. Set conditions: - SW: Writing 1 to one of these bits with the unsecure APB interface will clear the corresponding bit in **RSSIE.SNSSIEi** register.

3.3.2 Common Agent Function registers

3.3.2.1 Buffer pool function registers

(1) CABPIBWMCi (i=0..7)

Common Agent Buffer Pool IPV Based Watermark Configuration i.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV		IBSWMPNi[LCL_PTR_W1-1:0]													
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV		IBUWMPNi[LCL_PTR_W1-1:0]													

Bits	Bit name	RW-P	Initial value	Function description
31: LCL_PTR_W1+16	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_PTR_W1+15: 16	IBSWMPNi	RW-F	LCL_PTR_N - (8- i)*PORT_N*DEF_FRAME_SIZE/ LCL_RAM_BSZ - PORT_TSNA_N*6 - PORT_GWCA_N*3 - 1*3	IPV Based Secure Watermark Pointer Number i Functions: - For i equal to 7, Setting this register to LCL_PTR_N disables the IPV based Watermark function for IPV 7. - For i different than 7, Setting this register to **CABPIBWMC{i+1}.IBSWMPN{i+1}** disables the IPV based Watermark function for IPV i. - Refer to section 5.1.1 for details. Restrictions: - SW: This register shouldn't be set to a value greater than LCL_PTR_N. - SW: This register shouldn't be set to a value greater than **CABPIBWMC{i+1}.IBSWMPN{i+1}**
15: LCL_PTR_W1-1	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_PTR_W1-1:0	IBUWMPNi	RW-P	LCL_PTR_N - (8- i)*PORT_N*DEF_FRAME_SIZE/ LCL_RAM_BSZ - PORT_TSNA_N*6 - PORT_GWCA_N*3 - 1*3	IPV Based Unsecure Watermark Pointer Number i Functions: - For i equal to 7, Setting this register to LCL_PTR_N disables the IPV based Watermark function for IPV 7. - For i different than 7, Setting this register to **CABPIBWMC{i+1}.IBUWMPN{i+1}** disables the IPV based Watermark function for IPV i. - Refer to section 5.1.1 for details. Restrictions: - SW: This register shouldn't be set to a value greater than LCL_PTR_N. - SW: This register shouldn't be set to a value greater than **CABPIBWMC{i+1}.IBUWMPN{i+1}**

(2) CABPWMLC

Common Agent Buffer Pool Watermark Level Configuration

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV		WMCL[LCL_PTR_W1-1:0]													
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV		WMFL[LCL_PTR_W1-1:0]													

Bits	Bit name	R/W-P	Initial value	Function description
31: LCL_PTR_W1+16	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_PTR_W1+15 :16	WMCL	RW-P	LCL_PTR_N	Watermark critical level. Functions: - Setting this register to LCL_PTR_N disables the Watermark critical level function. - Refer to section 5.1.2 for details. Restrictions: - SW: This register value should always be smaller or equal to **CABPWMLC.WMFL.** - SW: This register shouldn't be set to a value greater than LCL_PTR_N.
15: LCL_PTR_W1	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_PTR_W1-1:0	WMFL	RW-P	LCL_PTR_N	Watermark flush level Function: - Setting this register to LCL_PTR_N disables the Watermark flush level function. - Refer to section 5.1.2 for details. Restrictions: - SW: This register shouldn't be set to a value greater than LCL_PTR_N.

(3) CABPPFLCi (i = 0..PAS_LVL_N -1)

Common Agent Buffer Pointer Pause Frame Level Configuration i.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV		PALi[LCL_PTR_W1-1:0]													
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV		PDLi[LCL_PTR_W1-1:0]													

Bits	Bit name	RW-P	Initial value	Function description
31: LCL_PTR_W1+16	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_PTR_W1+15 :16	PALi	RW-P	LCL_PTR_N	Pause Assertion Level Functions: - Setting this register to LCL_PTR_N will disable pause frames. - Refer to section 5.2.1 for details. Restrictions: - SW: This register shouldn't be set to a value greater than LCL_PTR_N. - SW: This register shouldn't be set to a value smaller than PORT_N*6.
15: LCL_PTR_W1	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_PTR_W1-1:0	PDLi	RW-P	LCL_PTR_N	Pause De-assertion Level Functions: - Refer to section 5.2.1 for details. Restrictions: - SW: This register value should always be smaller or equal to **CABPPFLCi.PALi.** - SW: This register shouldn't be set to a value greater than LCL_PTR_N. - SW: This register shouldn't be set to a value smaller than PORT_N*6. Precautions: - If **CABPPFLCi.PALi** value is too close to **CABPPFLCi.PDLi** value, corresponding **PAUSE signals** will change state rapidly and if zero automatic pause frames are activated [RMAC], it can result on a succession of pause frames on RMAC TX side which will decrease the link performance.

(4) CABPPWMLCi (i = 0..PORT_N-1)

Common Agent Buffer Pool per Port Watermark Level Configuration i.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV		PWMCLi[LCL_PTR_W1-1:0]													
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV		PWMFLi[LCL_PTR_W1-1:0]													

Bits	Bit name	RW-P	Initial value	Function description
31: LCL_PTR_W1+16	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_PTR_W1+15 :16	PWMCLi	RW-P	LCL_PTR_N	Per Port Watermark critical level i. Functions: - Setting this register to LCL_PTR_N disables the Watermark critical level function. - Refer to section 5.1.3 for details. Restrictions: - SW: This register value should always be smaller or equal to **CABPPWMLCi.PWMFLi.** - SW: This register shouldn't be set to a value greater than LCL_PTR_N.
15: LCL_PTR_W1	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_PTR_W1-1:0	PWMFLi	RW-P	LCL_PTR_N	Per Port Watermark flush level i Function: - Setting this register to LCL_PTR_N disables the Watermark flush level function. - Refer to section 5.1.3 for details. Restrictions: - SW: This register shouldn't be set to a value greater than LCL_PTR_N.

(5) CABPPPFLCij (i = 0..PORT_N -1) (j = 0..PAS_LVL_N-1)

Common Agent Buffer Pointer per Port Pause Frame Level Configuration i.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV		PPALij[LCL_PTR_W1-1:0]													
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV		PPDLij[LCL_PTR_W1-1:0]													

Bits	Bit name	RW-P	Initial value	Function description
31: LCL_PTR_W1+16	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_PTR_W1+15 :16	PPALij	RW-P	LCL_PTR_N	Per Port Pause Assertion Level ij. Functions: - Setting this register to LCL_PTR_N will disable pause frames. - Refer to section 5.2.2 for details. Restrictions: - SW: This register shouldn't be set to a value greater than LCL_PTR_N. - SW: This register shouldn't be set to a value smaller than 6.
15: LCL_PTR_W1	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_PTR_W1-1:0	PPDLij	RW-P	LCL_PTR_N	Per Port Pause De-assertion Level ij. Functions: - Refer to section 5.2.2 for details. Restrictions: - SW: This register value should always be smaller or equal to **CABPPPFLCij.PPALij.** - SW: This register shouldn't be set to a value greater than LCL_PTR_N. - SW: This register shouldn't be set to a value smaller than 6. Precautions: - If **CABPPPFLCij.PPALij** value is too close to **CABPPPFLCij.PPDLij** value, corresponding **PAUSE signals** will change state rapidly and if zero automatic pause frames are activated [RMAC], it can result on a succession of pause frames on RMAC TX side which will decrease the link performance.

(6) CABPULCi (i = 0..PORT_N-1)

Common Agent Buffer Pointer Utilization Level Configuration i.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV		MNNPNi[LCL_PTR_W1-1:0]													
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV		MXNPNi[LCL_PTR_W1-1:0]													

Bits	Bit name	RW-P	Initial value	Function description
31: LCL_PTR_W1+16	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_PTR_W1+15 :16	MNNPNi	RW-P	0H	Minimum Number of Pointer for port i. Minimum number of pointers that can be used by port i. Functions: - If **SUM(CABPULCi.MNNPNi)>=CABPPCM.RPC+SUM(CABPCPMi.RPCPi)** only ports for which **CABPCPMi.RPCPi<CABPULCi.MNNPNi** will be able to receive pointers. - Refer to section 5.3 for details. Restrictions: - SW: The sum of all **CABPULCi.MNNPNi** should be smaller or equal to LCL_PTR_N - SW: This register should always be smaller or equal to **CABPULCi.MXNPNi.**
15: LCL_PTR_W1	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_PTR_W1-1:0	MXNPNi	RW-P	LCL_PTR_N	Maximum Number of Pointer for port i Maximum number of pointers that can be used by port i. Functions: - If register **CABPCPMi.RPCPi** is equal to this register value, port i won't receive pointer anymore. - Refer to section 5.3 for details. Restrictions: - SW: The maximum number of pointers should be set to a value big enough to save 2 frames with a size equal to RMAC reception maximum frame size (set in RMAC [RMAC]).

Notes :
If CABPULC[i].MNNPN[i] of port[i] is changed(reconfigured) from its default value, then it restricts the reception maximum frame size of other ports(i.e. ports other than port[i]). The restricted reception maximum frame size of ports other than port[i] in such case is given by below expression,
Restricted reception maximum frame size = ((LCL_PTR_N – CABPULC[i].MNNPN[i])/(PORT_N - 1)) * 128 bytes. This is because other ports(ports other than port[i]) have to share remaining pointers(LCL_PTR_N – CABPULC[i].MNNPN[i]).
If this restriction is not followed, then the consumed buffer pointers will not be released by other ports and the frame will be stuck inside switch. Because the total buffer pointers are insufficient for the port in the middle of reception. To resolve this, reconfigure CABPULC[i].MNNPN[i] value to a smaller value.

(7) CABPIRM

Common Agent Buffer Pool Initialization Register Monitoring.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV														BPR	BPIOG

Bits	Bit name	RW-P	Initial value	Function description
31: 2	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
1	BPR	R-RP	0H	Buffer Pool Ready Set conditions: - When **CABPIRM.BPIOG** is getting cleared. - HW: This bit is set at "*clk_period[ns]*LCL_PTR_N*" time from buffer pool initialization starting. Clear condition - HW: If buffer pool is not ready.
0	BPIOG	R!=W-RP	0H	Buffer Pool Initialization Ongoing Set conditions: - SW: By writing 1 to this register. Clear condition: - HW: This bit is cleared when buffer pool initialization is finished. Restriction: - HW: CPU cannot write one to this bit when **CABPIRM.BPR** is set. Security restriction: - HW: For write access, this register cannot be accessed by the unsecure APB interface.

(8) CABPPCM

Common Agent Buffer Pool Pointer Count Monitoring.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV		TPC[LCL_PTR_W1-1:0]													
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV		RPC[LCL_PTR_W1-1:0]													

Bits	Bit name	RW-P	Initial value	Function description
31: LCL_PTR_W1+16	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_PTR_W1+15 :16	TPC	R-P	LCL_PTR_N	Total pointer count. Functions: - Read LCL_PTR_N value, fixed to LCL_PTR_N value. - Reading this register gives the number of pointers contained in the buffer pool. The switch local data RAM size will be equal to LCL_RAM_BSZ*__CABPPCM.TPC__ bytes.
15: LCL_PTR_W1	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_PTR_W1-1:0	RPC	R-P	LCL_PTR_N	Remaining pointer count. Increment conditions: - HW: Incremented by 1 when a pointer is released. Decrement conditions: - HW: Decremented by 1 when a pointer is given to an agent Restrictions: - HW: if __CAEIS0.PECCES__ has been set, there could be a deviation between this counter and the real number of pointers in the buffer pool.

(9) CABPLCM

Common Agent Buffer Pool Pointer Least Count Monitoring.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV		LRC[LCL_PTR_W1-1:0]													

Bits	Bit name	RW-P	Initial value	Function description
31: LCL_PTR_W1	RSV	R0-U	0H	Reserved area. On read, 0s will be returned
LCL_PTR_W1-1:0	LRC	RC-P	LCL_PTR_N	Least remaining pointer count. This register remembers the smaller value that has been taken by the pointer remaining counter. Update condition: - HW: This register is updated to **CABPPCM.RPC** when **CABPPCM.RPC**<**CABPLCM.LRC** Clear conditions: - Reading this register clears it to **CABPPCM.RPC** value**.**

(10) CABPCPMi (i = 0..PORT_N-1)

Common Agent Buffer Pointer Count per Port Monitoring i.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV		RPCPi[LCL_PTR_W1-1:0]													

Bits	Bit name	RW-P	Initial value	Function description
31:LCL_PTR_W1	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_PTR_W1-1:0	RPCPi	R-P	0H	Received pointer count for Port i. Functions: - This register shows the number of pointers that are used by agent i to store data. Increment conditions: - HW: Incremented by 1 when a pointer is given to port i. Decrement conditions: - HW: Decremented by 1 when a pointer which has been given to port i is being released. Restrictions: - HW: if **CAEIS0.PECCES** has been set, there could be a deviation between this counter and the real number of pointers in the buffer pool.

(11) CABPMCPMi (i = 0..PORT_N-1)

Common Agent Buffer Pointer Maximum Count per Port Monitoring i.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV		RPMCPi[LCL_PTR_W1-1:0]													

Bits	Bit name	RW-P	Initial value	Function description
31:LCL_PTR_W1	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_PTR_W1-1:0	RPMCPi	RC-P	0H	Received pointer Maximum count for Port i. Functions: - This register remembers the biggest value than has been taken by **CABPMCPMi.RPCPi** since last read access. Update condition: - HW: This register is updated to **CABPMCPMi.RPCPi** when **CABPMCPMi.RPCPi** >**CABPMCPMi.RPMCPi** Clear conditions: - Reading this register clears it to **CABPMCPMi.RPCPi** value**.** Restrictions: - It may momentarily exceed several upper and lower limit settings.

3.3.2.2 Reject RAM function registers

(1) CARDNM

Common Agent Rejected Descriptor Number Monitoring.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV		RDNRR[LCL_PTR_W1-1:0]													

Bits	Bit name	RW-P	Initial value	Function description
31: LCL_PTR_W1	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_PTR_W1-1:0	RDNRR	R-P	0B	Rejected Descriptor Number in Reject RAM Functions: - This register shows the number of descriptors currently contained in the reject RAM. Increment conditions: - HW: Incremented by 1 a descriptor is received from Forwarding Engine. Decrement conditions: - HW: Decremented by 1 when a descriptor is read from reject RAM.

(2) CARDMNM

Common Agent Rejected Descriptor Maximum Number Monitoring.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV		RDMNRR[LCL_PTR_W1-1:0]													

Bits	**Bit name**	**RW-P**	**Initial value**	**Function description**
31: LCL_PTR_W1	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_PTR_W1-1:0	RDMNRR	RC-P	0B	Rejected Descriptor Maximum Number in Reject RAM Functions: - This register remembers the biggest value than has been taken by **CARDNM.RDNRR** since last read access. Update conditions: - Set to **CARDNM.RDNRR** when **CARDNM.RDNRR> CARDMNM.RDMNRR**. Clear conditions: - Reading this register clears it to **CARDNM.RDNRR** value**.**

3.3.3 Common Agent Counter registers

This section describes Common Agent counter registers.

3.3.3.1 Reject RAM counter registers

(1) CARDCN

Common Agent Rejected Descriptor CouNter

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDN[COUNT_MED_W-1:0]															

Bits	Bit name	RW-P	Initial value	Function description
31: COUNT_MED_W	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
COUNT_MED_W-1:0	RDN	RC-P	0H	Rejected Descriptor Number Functions: - This register counts the number of descriptors that has been rejected since the last read. Clear conditions: - SW: Reading this register clears it. Increment conditions: - HW: Incremented by 1 when a rejected descriptor is received “under the following conditions” and if this register has a value different than {{COUNT_MED_W}{1’b1}}. - [FWD] Any filtering. - [GWCA] Descriptor queue overflow error or Descriptor queue security error. Even if errors are detected on multiple ports ([GWCA] or [TSNA]) at the same time, there is only one original descriptor, so this counts only once. - [TSNA] Descriptor queue overflow error or Descriptor queue security error. Even if errors are detected on multiple ports ([GWCA] or [TSNA]) at the same time, there is only one original descriptor, so this counts only once.

3.3.4 Common Agent Interrupt registers

This section describes Common Agent interrupt registers.

3.3.4.1 Error interrupt registers

This subsubsection describes Common Agent error interrupts.

(1) CAEIS0

Common Agent Error Interrupt Status 0.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV												EEIPLN [LCL_RAM_ECCR_W1-1:0]			
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV					WMFLOS	WMCLOS	BPOPS	RSV					BPECCES	DSECCES	PECCES

Bits	Bit name	RW-P	Initial value	Function description
31: LCL_RAM_ECCR_W1+16	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
LCL_RAM_ECCR_W1+15 :16	EEIPLN	R-P	0H	ECC Error Inducing Pointer Loss Number Functions: - This register shows the number of RAM addresses that has been blocked because of ECC errors. - The buffer pool lost between **CAEIS0.EEIPLN** and **CAEIS0.EEIPLN***2 pointers because of **CAEIS0.BPECCES** ECC error. Increment conditions: - HW: Incremented by 1 if an ECC error is received for buffer pool RAM on an address where ECC error didn't happened before and if **CAEIS0.EEIPLN** is different than LCL_RAM_ECCR_N.
15: 11	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
10	WMFLOS	R!=W-P	0H	Watermark Flush Level Overtook Status. Set conditions: - HW: When watermark flush level **WM.FLUSH** gets asserted to All1 because of global level-based watermark (Refer to Fig 5.2). Clear conditions: - SW: Writing 1 to this bit will clear it. Error recovery: - SW: No recovery, data can be lost [FWD].

9	WMCLOS	R!=W-P	0H	Watermark Critical Level Overtook Status. Set conditions: - HW: When watermark critical level **WM.CRITICAL** gets asserted to All1 because of global level-based watermark (Refer to Fig 5.2). Clear conditions: - SW: Writing 1 to this bit will clear it. Error recovery: - SW: No recovery, data can be lost [FWD]
8	BPOPS	R!=W-P	0H	Buffer Pool Out of Pointer Status Set conditions: - HW: when **CABPPCM.RPC** becomes null. Clear conditions: - SW: Writing 1 to this bit will clear it. Error recovery: - SW: No recovery, switch could overflow [RMAC] When using Cut through, please use the Watermark function so that this status does not occur. If get this error (it may get hung-up in the transmit state[TSNA and RMAC]), initialize the state with Emergency reset flow [TOP].
7:3	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
2	BPECCES	R!=W-P	0H	Buffer Pool ECC error interrupt status. Set conditions: - HW: When an ECC error has been detected in the buffer pool. Clear conditions: - SW: Writing 1 to this bit will clear it. Error recovery: - HW: The address access will be blocked to allow operation continuation. The buffer pool is able to block until LCL_RAM_ECCR_N addresses. Anytime an address is blocked, it will be counted in **CAEIS0.EEIPLN** register. - SW: if **CAEIS0.EEIPLN**<LCL_RAM_ECCR_N: Nothing if it is acceptable to operate with a reduced Local RAM else Switch reset. - SW: if **CAEIS0.EEIPLN**=LCL_RAM_ECCR_N: Switch reset.
1	DSECCES	R!=W-P	0H	Descriptor ECC error interrupt status. Set conditions: - HW: When an ECC error has been detected while reading a descriptor from the reject RAM. Clear conditions: - SW: Writing 1 to this bit will clear it. Error recovery: - HW: Lost of the error descriptor. One or several pointers are lost but the switch will continue operating normally with a reduced Local RAM. - SW: Nothing if it is acceptable to operate with a reduced Local RAM else Switch reset.

0	PECCES	R!=W-P	0H	Pointer ECC error interrupt status. Set conditions: - HW: When an ECC error has been detected in a pointer from the fabric read interface while rejecting a frame received from the forwarding engine Descriptor Reject interface. Clear conditions: - SW: Writing 1 to this bit will clear it. Error recovery: - HW: One or several pointers are lost but the switch will continue operating normally with a reduced Local RAM. - SW: Nothing if it is acceptable to operate with a reduced Local RAM else Switch reset.

(2) CAEIE0

Common Agent Error Interrupt Enable0.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV					WMFLOE	WMCLOE	BPOPE	RSV					BPECCEE	DSECCEE	PECCEE

Bits	Bit name	RW-P	Initial value	Function description
31:11	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
10	WMFLOE	R!=W-P	0H	Watermark Flush Level Overtook Enable. Set conditions: - Writing 1 to this bit will set it. Clear conditions: - Writing 1 to **CAEID0.WMFLOD** register will clear this bit.
9	WMCLOE	R!=W-P	0H	Watermark critical level overtook Enable. Set conditions: - Writing 1 to this bit will set it. Clear conditions: - Writing 1 to **CAEID0.WMCLOD** register will clear this bit.
8	BPOPE	R!=W-P	0H	Buffer Pool Out of Pointer Enable. Set conditions: - Writing 1 to this bit will set it. Clear conditions: - Writing 1 to **CAEID0.BPOPD** register will clear this bit.
7:3	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
2	BPECCEE	R!=W-P	0H	Buffer Pool ECC error interrupt Enable. - 1'b0: Interrupt disabled. - 1'b1: Interrupt Enabled. Set conditions: - Writing 1 to this bit will set it. Clear conditions: - Writing 1 to **CAEID0.BPECCED** register will clear this bit.
1	DSECCEE	R!=W-P	0H	Descriptor ECC error interrupt Enable. - 1'b0: Interrupt disabled. - 1'b1: Interrupt Enabled. Set conditions: - Writing 1 to this bit will set it. Clear conditions: - Writing 1 to **CAEID0.DSECCED** register will clear this bit.

0	PECCEE	R!=W-P	0H	Pointer ECC error interrupt Enable. - 1'b0: Interrupt disabled. - 1'b1: Interrupt Enabled. Set conditions: - Writing 1 to this bit will set it. Clear conditions: - Writing 1 to **CAEID0.PECCED** register will clear this bit.

(3) CAEID0

Common Agent Error Interrupt Disable 0.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV					WMFLOD	WMCLOD	BPOPD	RSV					BPECCED	DSECCED	PECCED

Bits	Bit name	RW-P	Initial value	Function description
31:11	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
10	WMFLOD	R0W-P	0H	Watermark Flush Level Overtook Disable. Function: - Writing 1 to this bit will clear **CAEIE0.WMFLOE** register.
9	WMCLOD	R0W-P	0H	Watermark critical level overtook Disable. Function: - Writing 1 to this bit will clear **CAEIE0.WMCLOE** register.
8	BPOPD	R0W-P	0H	Buffer Pool Out of Pointer Disable Function: - Writing 1 to this bit will clear **CAEIE0.BPOPE** register.
7:3	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
2	BPECCED	R0W-P	0H	Buffer Pool ECC error interrupt Disable. Function: - Writing 1 to this bit will clear **CAEIE0.BPECCEE** register.
1	DSECCED	R0W-P	0H	Descriptor ECC error interrupt disable. Function: - Writing 1 to this bit will clear **CAEIE0.DECCEE** register.
0	PECCED	R0W-P	0H	Pointer ECC error interrupt Disable. Function: - Writing 1 to this bit will clear **CAEIE0.PECCEE** register.

(4) CAEIS1

Common Agent Error Interrupt Status 1.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV / PWMFLOS[PORT_N-1:0]															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV / PWMCLOS[PORT_N-1:0]															

Bits	**Bit name**	**RW-P**	**Initial value**	**Function description**
31: PORT_N+16	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
PORT_N+15:16	PWMFLOS	R!=W-P	0H	Per Port Watermark Flush Level Overtook Status. Set condition: - HW: For bit i, when watermark flush level bit i **WM.FLUSH[i]** gets asserted to 1'b1 because of per-Port level-based watermark (Refer to Fig 5.3). Clear conditions: - SW: Writing 1 to this one of these bits will clear it. Error recovery: - No recovery, data can be lost [FWD].
15:PORT_N	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
PORT_N-1:0	PWMCLOS	R!=W-P	0H	Per Port Watermark Critical Level Overtook Status. Set condition: - HW: For bit i, when watermark critical level bit i **WM.CRITICAL[i]** gets asserted to 1'b1 because of per-Port level-based watermark (Refer to Fig 5.3). Clear conditions: - SW: Writing 1 to this one of these bits will clear it. Error recovery: - No recovery, data can be lost [FWD]

(5) CAEIE1

Common Agent Error Interrupt Enable 1.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV / PWMFLOE[PORT_N-1:0]															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV / PWMCLOE[PORT_N-1:0]															

Bits	**Bit name**	**RW-P**	**Initial value**	**Function description**
31: PORT_N+16	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
PORT_N+15:16	PWMFLOE	R!=W-P	0H	Per Port Watermark Flush Level Overtook Enable. Set condition: - Writing 1 to one of these bits will set it. Clear conditions: - SW: Writing 1 to one of **CAEID1.PWMFLOD** bits will clear corresponding in this register.
15:PORT_N	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
PORT_N-1:0	PWMCLOE	R!=W-P	0H	Per Port Watermark Critical Level Overtook Enable. Set condition: - Writing 1 to one of these bits will set it. Clear conditions: - SW: Writing 1 to one of **CAEID1.PWMCLOD** bits will clear corresponding in this register.

(6) CAEID1

Common Agent Error Interrupt Disable 1.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV / PWMFLOD[PORT_N-1:0]															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV / PWMCLOD[PORT_N-1:0]															

Bits	**Bit name**	**RW-P**	**Initial value**	**Function description**
31: PORT_N+16	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
PORT_N+15:16	PWMFLOD	R0W-P	0H	Per Port Watermark Flush Level Overtook Disable. Functions: - SW: Writing 1 to one of these bits will clear corresponding **CAEIE1.PWMFLOE** bit.
15:PORT_N	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
PORT_N-1:0	PWMCLOD	R0W-P	0H	Per Port Watermark Critical Level Overtook Disable. Functions: - SW: Writing 1 to one of these bits will clear corresponding **CAEIE1.PWMCLOE** bit.

3.3.4.2 Monitoring interrupt registers

This subsubsection describes Common Agent monitoring interrupts.

(1) CAMIS0

Common Agent Monitoring Interrupt Status 0.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV														PFS[PAS_LVL_N-1:0]	

Bits	Bit name	RW-P	Initial value	Function description
31: PAS_LVL_N	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
PAS_LVL_N-1:0	PFS	R!=W-P	0H	Pause Frame Status. Set conditions: - HW: For bit i, when Pause Frame i starts to happen (pause assertion timing). Refer to section 5.2. Clear conditions: - SW: Writing 1 to this one of these bits will clear it.

(2) CAMIE0

Common Agent Monitoring Interrupt Enable 0.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV														PFE[PAS_LVL_N-1:0]	

Bits	**Bit name**	**RW-P**	**Initial value**	**Function description**
31: PAS_LVL_N	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
PAS_LVL_N-1:0	PFE	R!=W-P	0H	Pause Frame Enable. Set conditions: - Writing 1 to one of these bits will set it. Clear conditions: - SW: Writing 1 to one of **CAMID0.PFD** bits will clear corresponding in this register.

(3) CAMID0

Common Agent Monitoring Interrupt Disable 0.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV														PFD[PAS_LVL_N-1:0]	

Bits	Bit name	RW-P	Initial value	Function description
31: PAS_LVL_N	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
PAS_LVL_N-1:0	PFD	R0W-P	0H	Pause Frame Disable. Function: - SW: Writing 1 to one of these bits will clear corresponding **CAMIE0.PFE** bit.

(4) CAMIS1

Common Agent Monitoring Interrupt Status 1.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV / PPFS[PORT_ N*PAS_LVL_N-1:0]															

Bits	Bit name	RW-P	Initial value	Function description
31:PORT_N* PAS_LVL_N	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
PORT_N* PAS_LVL_N-1:0	PPFS	R!=W-P	0H	Per Port Pause Frame Status. Set condition: - HW: For bit i, when Pause Frame i starts to happen (pause assertion timing). Refer to section 5.2. Pauser Frame i = Port “Integer of i/PAS_LVL_N” Pause Level “Remainder of i/PAS_LVL_N”. For example, bit 0 = Port 0 Pause Level 0, bit 1 = Port 0 Pause Level 1, bit 2 = Port 1 Pause Level 0… Clear conditions: - SW: Writing 1 to this one of these bits will clear it.

(5) CAMIE1

Common Agent Monitoring Interrupt Enable 1.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV / PPFE[PORT_N* PAS_LVL_N-1:0]															

Bits	Bit name	RW-P	Initial value	Function description
31:PORT_N* PAS_LVL_N	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
PORT_N* PAS_LVL_N-1:0	PPFE	R!=W-P	0H	Per Port Pause Frame Enable. Set conditions: - Writing 1 to one of these bits will set it. Clear conditions: - SW: Writing 1 to one of **CAMID1.PPFD** bits will clear corresponding in this register.

(6) CAMID1

Common Agent Monitoring Interrupt Disable 1.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV / PPFD[PORT_N* PAS_LVL_N-1:0]															

Bits	Bit name	RW-P	Initial value	Function description
31:PORT_N* PAS_LVL_N	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
PORT_N* PAS_LVL_N-1:0	PPFD	R0W-P	0H	Per Port Pause Frame Disable. Function: - SW: Writing 1 to one of these bits will clear corresponding **CAMIE1.PPFE** bit.

3.3.5 Common Agent Security registers

This section describes Common Agent security registers.

Security restrictions:

- The registers described in this section cannot be accessed by the unsecure APB interface.

(1) CASCR

Common Agent Security Configuration Register.

B31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RSV															
B15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSV										MIRSL	EIRSL	CRSL	RRRSL	BPRSL	VRSL

Bits	Bit name	RW-P	Initial value	Function description
31:6	RSV	R0-U	0H	Reserved area. On read, 0 will be returned.
5	MIRSL	RW-F	0H	Monitoring Interrupt Register Security Level Counter Register Security Level Values: - 1'b0: Monitoring interrupt registers can only be accessed by the APB secure interface - 1'b1: Monitoring interrupt registers can be accessed by both APBs Monitoring interrupt registers include the following registers: - **CAMIS0** - **CAMIE0** - **CAMID0** - **CAMIS1** - **CAMIE1** - **CAMID1**
4	EIRSL	RW-F	0H	Error Interrupt Register Security Level Counter Register Security Level Values: - 1'b0: Error interrupt registers can only be accessed by the APB secure interface - 1'b1: Error interrupt registers can be accessed by both APBs Error interrupt registers include the following registers: - **CAEIS0** - **CAEIE0** - **CAEID0** - **CAEIS1** - **CAEIE1** - **CAEID1**
3	CRSL	RW-F	0H	Counter Register Security Level Values: - 1'b0: Counter registers can only be accessed by the APB secure interface - 1'b1: Counter registers can be accessed by both APBs Counter registers include the following registers: - **CARDCN**

2	RRRSL	RW-F	0H	Reject RAM Register Security Level Values: - 1'b0: Reject RAM registers can only be accessed by the APB secure interface - 1'b1: Reject RAM registers can be accessed by both APBs Reject RAM registers include the following registers: - **CARDNM** - **CARDMNM**
1	BPRSL	RW-F	0H	Buffer Pool Register Security Level Values: - 1'b0: Buffer Pool registers can only be accessed by the APB secure interface - 1'b1: Buffer Pool registers can be accessed by both APBs Buffer Pool registers include the following registers: - **CABPIBWMCi** - **CABPWMLC** - **CABPPFLCi** - **CABPPWMLCi** - **CABPPPFLCij** - **CABPULCi** - **CABPIRM** - **CABPPCM** - **CABPLCM** - **CABPCPMi** - **CABPMCPMi**
0	VRSL	RW-F	0H	Version Register Security Level. Values: - 1'b0: Version registers can only be accessed by the APB secure interface - 1'b1: Version registers can be accessed by both APBs Version registers include the following registers: - **RIPV**

4. Register utilization

4.1 Software flows

Restrictions:

SW: Please follow the flow in this section.

4.1.1 Software flow legend

Software flow legend is described in Fig 4.1.

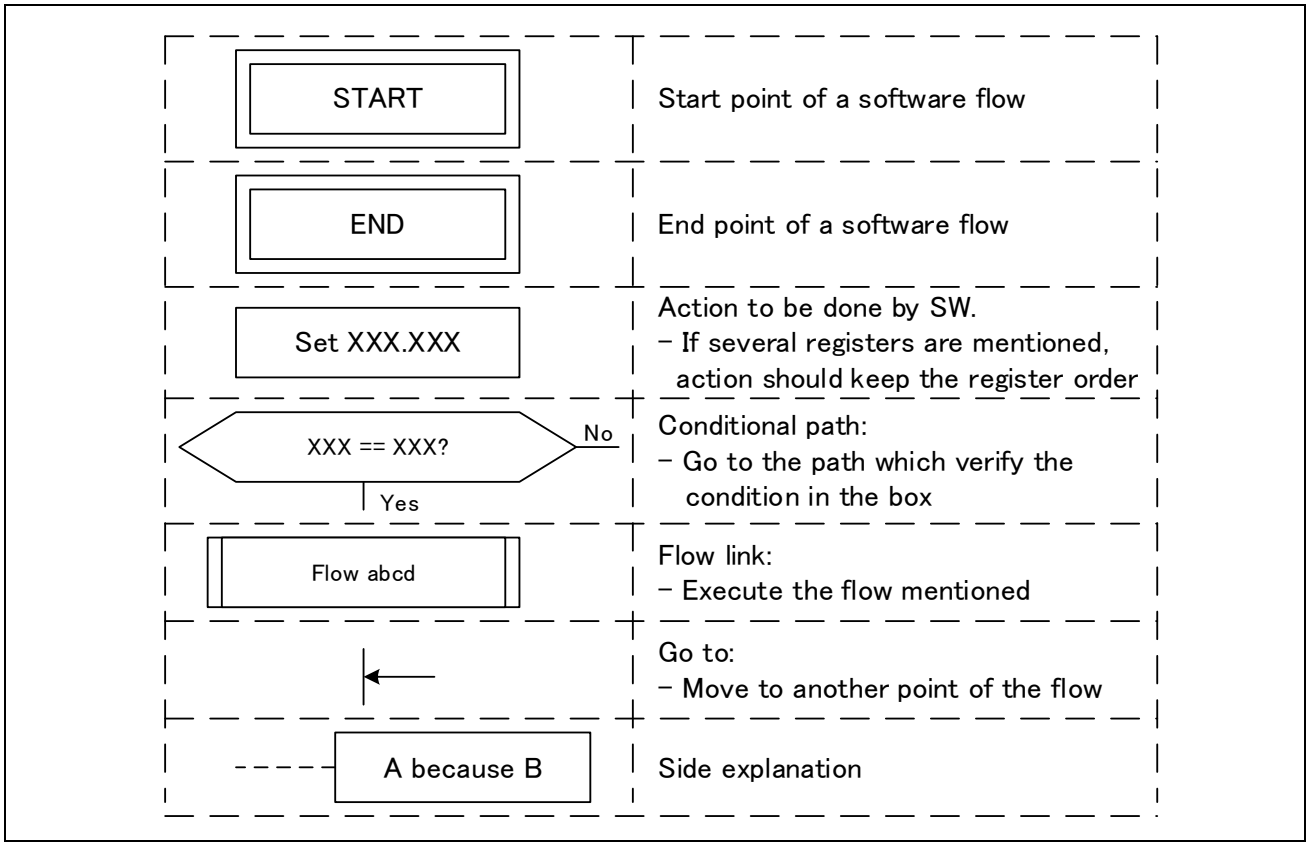


Fig 4.1: Software flow legend

4.1.2 Interrupt handling flow

Interrupt handling flow can be used in any mode except RESET mode.

The interrupt handling flow is described in Fig 4.2.

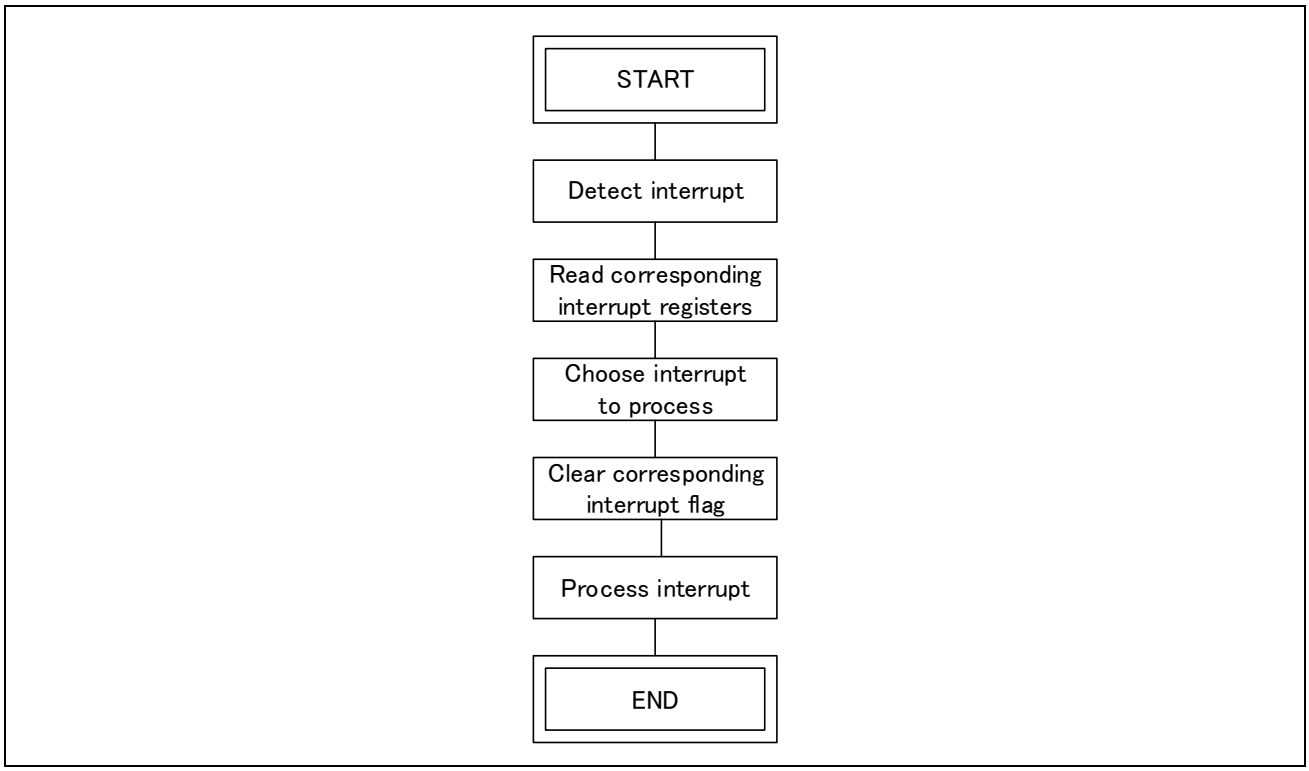


Fig 4.2: Interrupt handling flow

4.1.3 Called software flows

The flows described in this section can only be called from other flows thanks to a “flow link” box (Fig 4.1) and cannot be used alone.

4.1.3.1 Buffer pool initialization flow

The Buffer pool initialization flow is described in Fig 4.3.

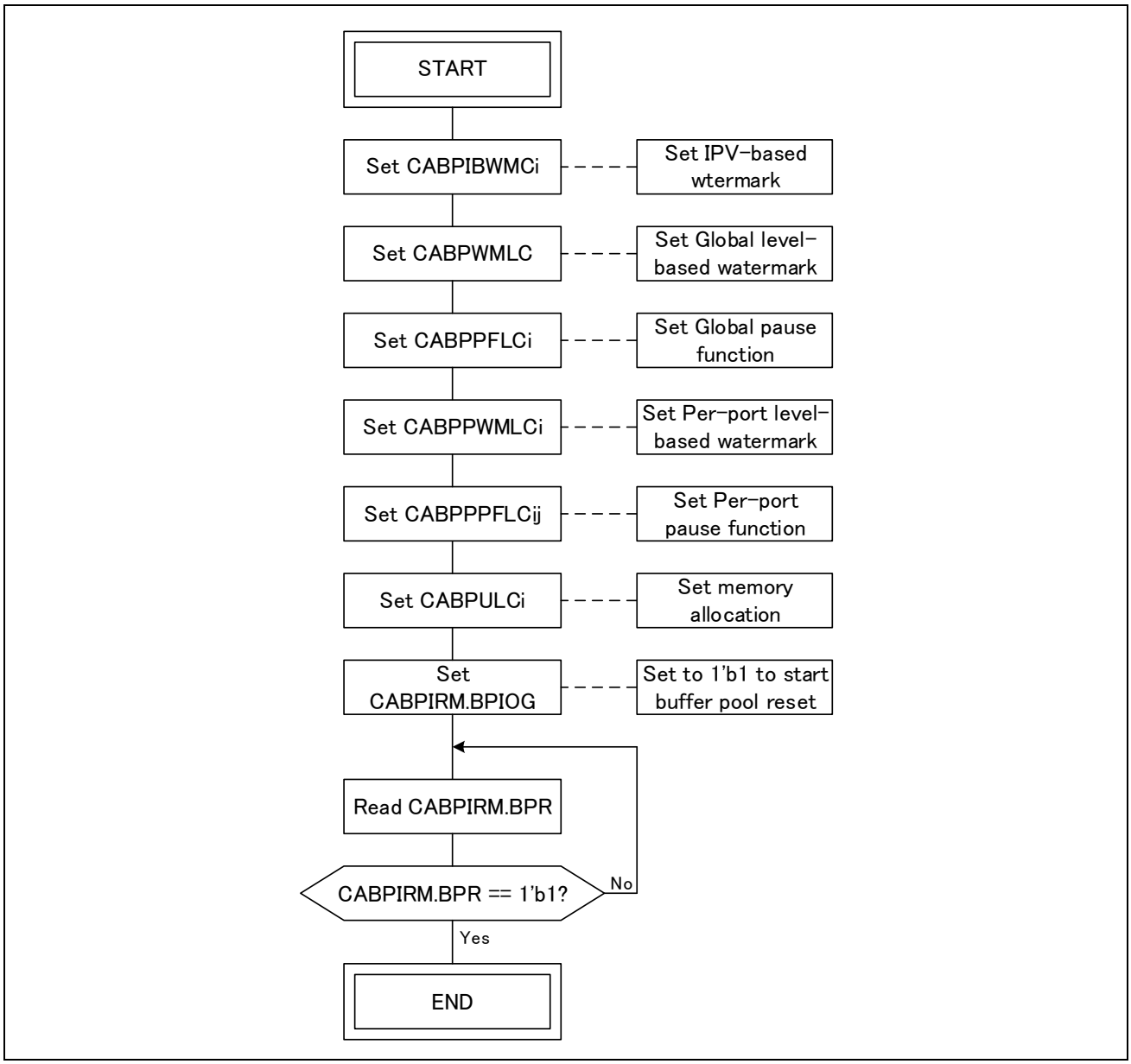


Fig 4.3: Buffer pool initialization flow

4.1.3.2 R-Switch reset flow

The R-Switch reset flow is described in Fig 4.4.

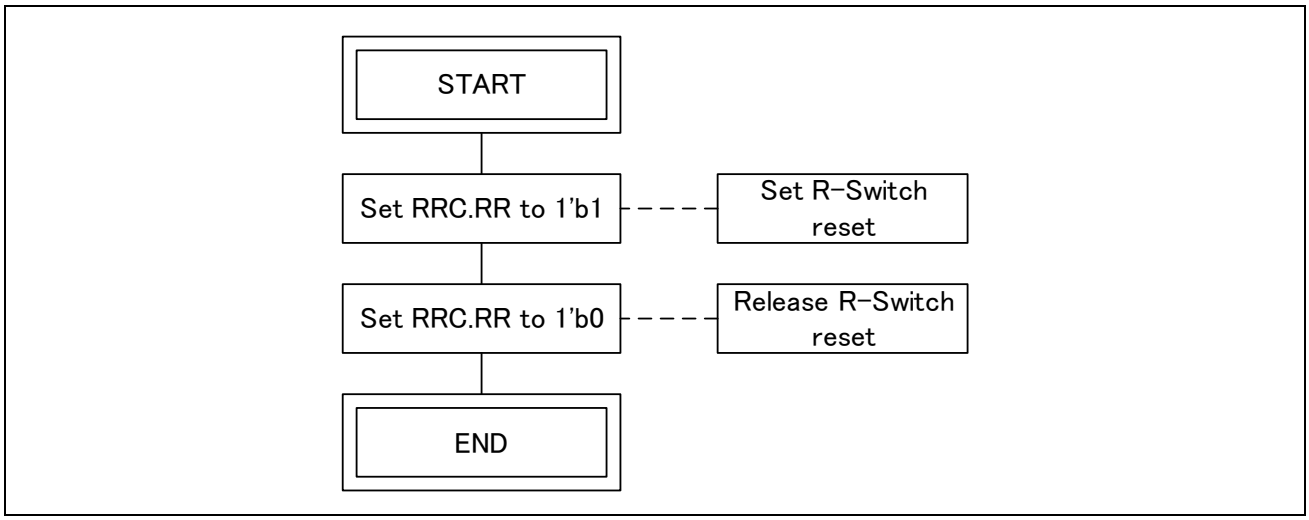


Fig 4.4: R-Switch reset flow

4.1.3.3 Agent i clock control flows (i = 0..PORT_N-1)

Agent i clock flows are described in Fig 4.5, Fig 4.6 and Fig 4.7.

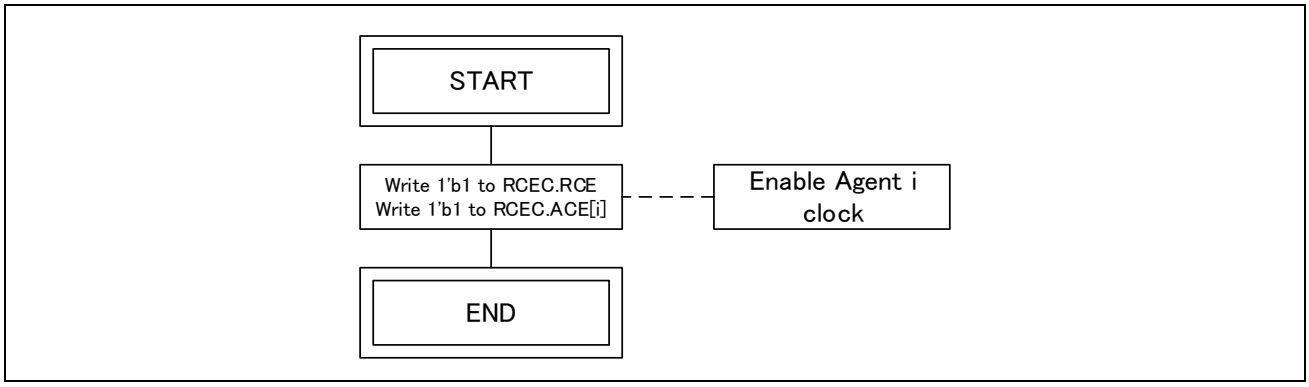


Fig 4.5: Agent i clock enable flow

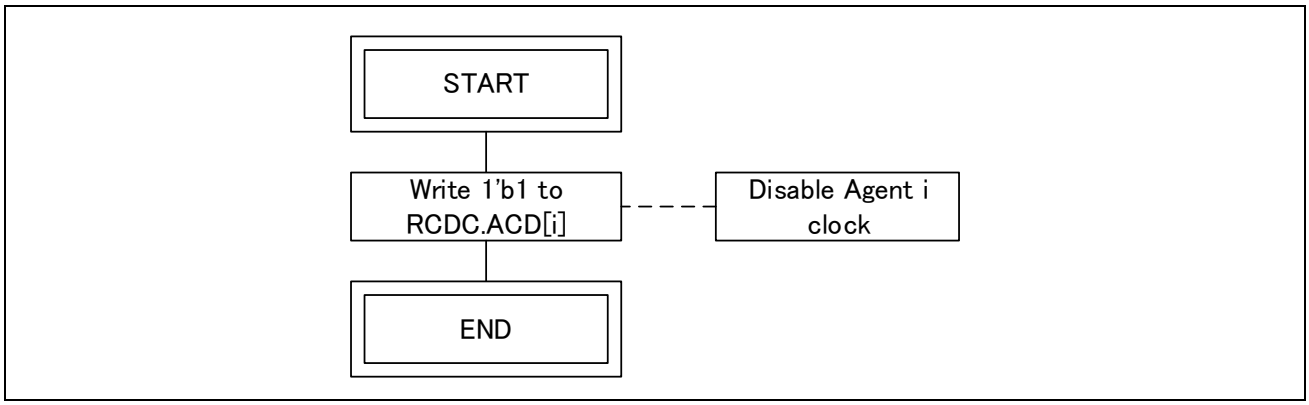


Fig 4.6: Agent i clock disable flow

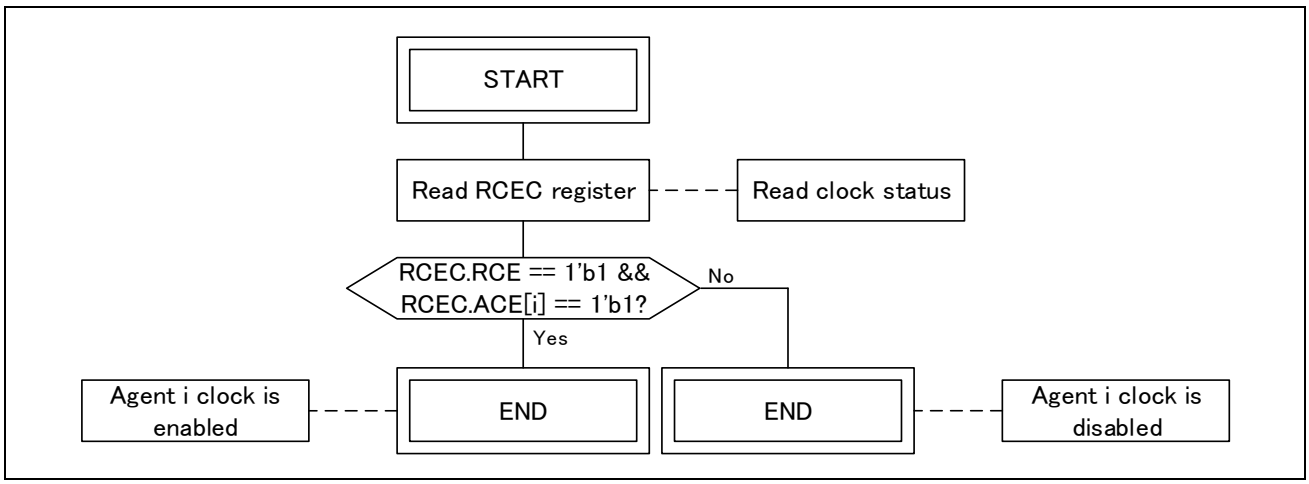


Fig 4.7: Agent i clock status check flow

4.1.3.4 Switch clock control flows

Switch clock flows are described in Fig 4.8 and Fig 4.9.

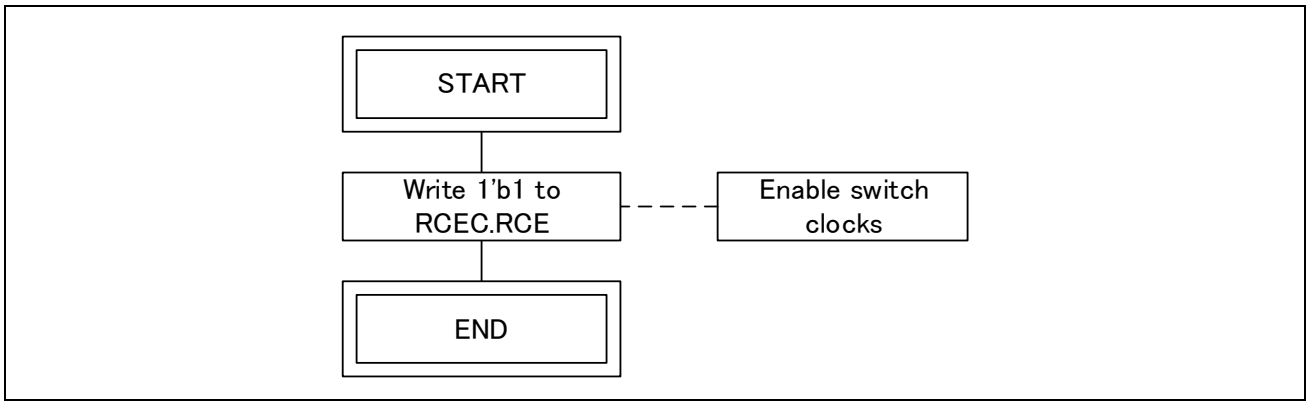


Fig 4.8: Switch clock enable flow

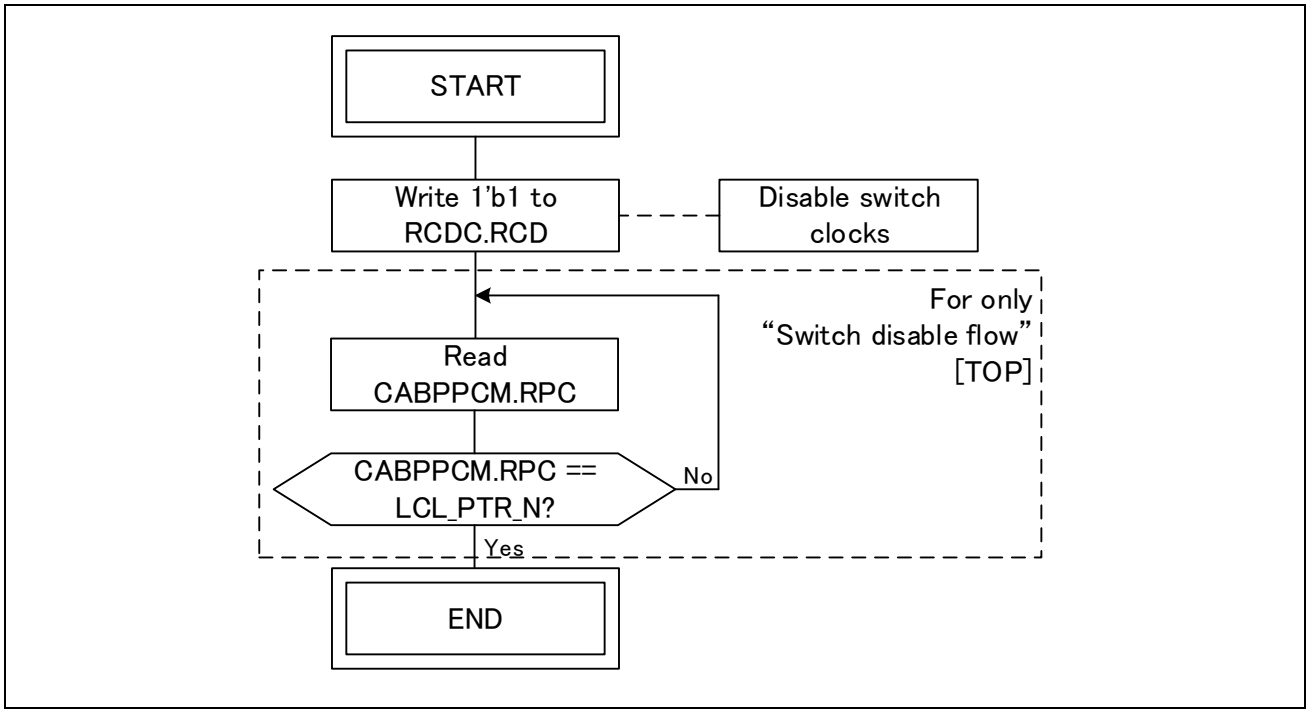


Fig 4.9: Switch clock disable flow

4.1.4 Register writable without software flow

These registers can be changed dynamically. (However, it is necessary that the initial settings such as the clock enabling have been completed.)

- CABPPFLCi
- CABPPPFLCij
- Other (not been described so far)

5. Functional details

5.1 Watermark function

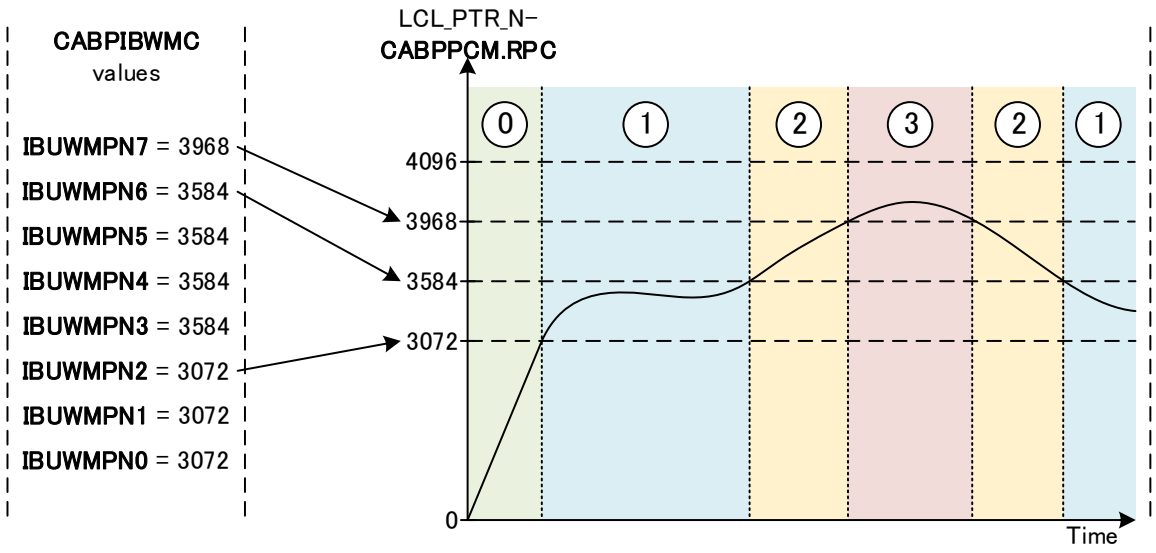
If a switch overflow occurs, frames in the switch will be loss at the ingress part of agents [GWCA] [TSNA]. It will result in a random loss of data independently from the actual priority of frames. To avoid so, the watermark function allows to selectively reject frames in forwarding engine [FWD] before a switch overflow occurs.

Common agent includes three watermark functions, the IPV based watermark, the global level-based watermark and the per-Port level-based watermark. The watermark functions aim at controlling the switch behavior in case of overflow by generating data loss. All watermark functions are running in parallel and a frame will be rejected if one or more of the watermark functions decide so.

5.1.1 IPV based watermark

The IPV based watermark ensure the minimum QoS requirement for a switch. It is enabled by the default configurations for a switch maximum frame size of DEF_FRAME_SIZE bytes. This function is controlled by the Common Agent, however, the data loss is performed by the Forwarding Engine [FWD]. The combination with other functions, "IPV based watermark" is affected by "Queue pause" regardless of "Frame size(DEF_FRAME_SIZE)".

Fig 5.1 gives an example for IPV based unsecure watermark setting for and LCL_PTR_N == 4096 (The IPV based secure watermark works the same way but using **CABPIBWMCi.IBSWMPNi** registers). Note that LCL_PTR_N-**CABPPCM.RPC** variations in Fig 5.1 are arbitrary and do not correspond to anything relevant.



	Explanation
0	No descriptor is rejected because **CABPIBWMC.IBUWMPN2** is bigger than LCL_PTR_N-**CABPPCM.RPC**
1	Descriptors with **FDESCR.IPV** 0, 1 or 2 are rejected because **CABPIBWMC.IBUWMPN2** is smaller than LCL_PTR_N-**CABPPCM.RPC**
2	Descriptors with **FDESCR.IPV** 0, 1, 2, 3, 4, 5 or 6 are rejected because **CABPIBWMC.IBUWMPN6** is smaller than LCL_PTR_N-**CABPPCM.RPC**
3	All descriptor are rejected because **CABPIBWMC.IBUWMPN7** is smaller than LCL_PTR_N-**CABPPCM.RPC**

Fig 5.1: Example for IPV based watermark setting (LCL_PTR_N == 4096)

Note:

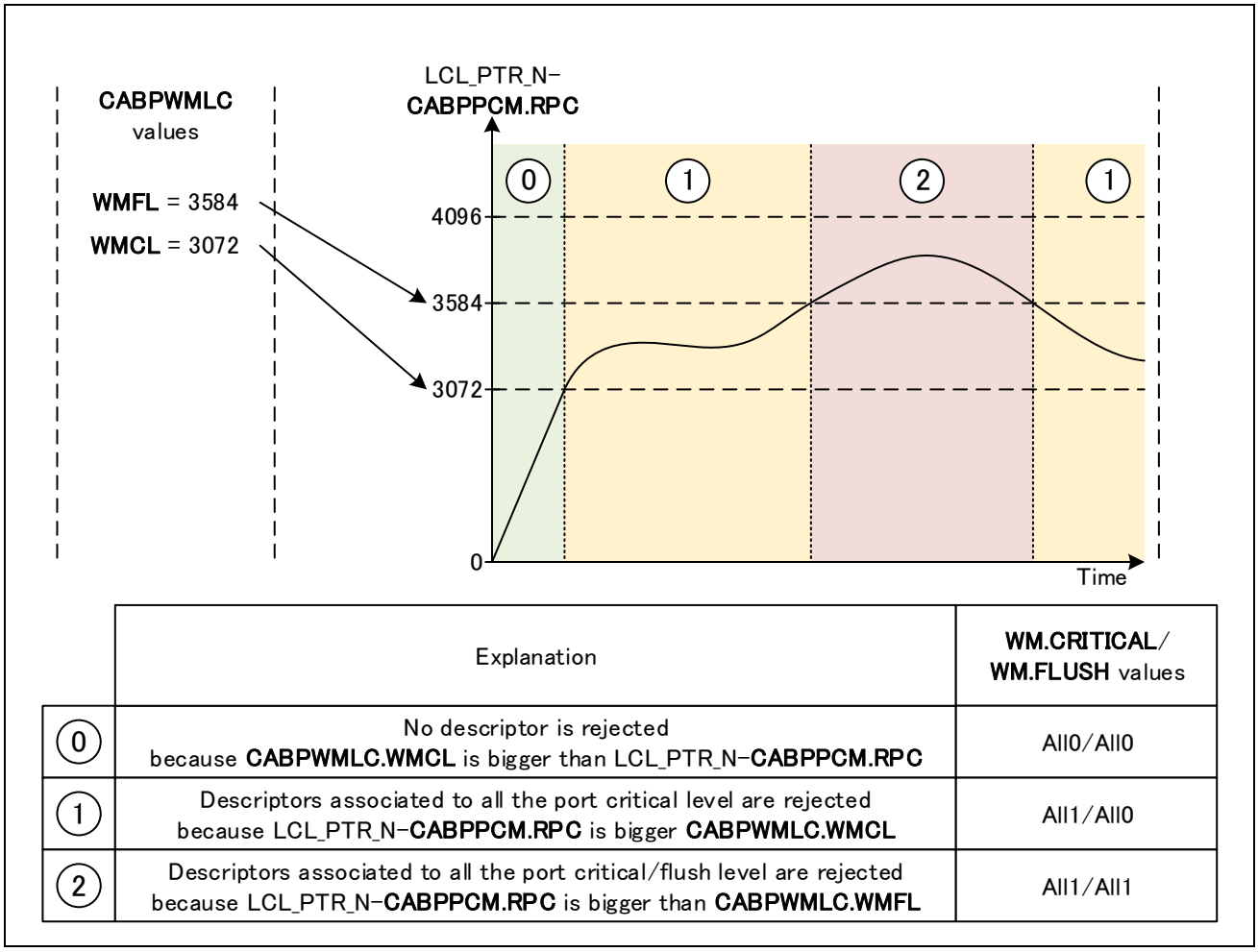
- **FDESCR.IPV** refers to the Internal Priority Value given to a frame by forwarding engine in the forwarding descriptor but before filtering. for more details, refer to forwarding engine specification document [FWD].

5.1.2 Global level-based watermark

The Global level-based watermark aims at rejecting frames with targeted IPVs depending on the number of pointers are currently used by the switch (**CABPPCM.RPC** register). This function is controlled by the Common Agent (through **WM.CRITICAL[PORT_N-1:0]** and **WM.FLUSH[PORT_N-1:0]**), however, the data loss is performed by the Forwarding Engine [FWD]. Note that **WM.CRITICAL** and **WM.FLUSH** interfaces can also be set by the per-Port level-based watermark (one of these pins will be set if the per-Port level-based watermark OR the global level-based watermark is in the required conditions to set it).

The Global level-based watermark use **CABPWMLC.WMFL** and **CABPWMLC.WMCL** registers**.**

Fig 5.2 gives an example for Global Level-based watermark setting for and LCL_PTR_N == 4096. Note that LCL_PTR_N-**CABPPCM.RPC** variations in Fig 5.2 are arbitrary and do not correspond to anything relevant.



	Explanation	WM.CRITICAL/ WM.FLUSH values
⓪	No descriptor is rejected because **CABPWMLC.WMCL** is bigger than LCL_PTR_N−**CABPPCM.RPC**	All0/All0
①	Descriptors associated to all the port critical level are rejected because LCL_PTR_N−**CABPPCM.RPC** is bigger **CABPWMLC.WMCL**	All1/All0
②	Descriptors associated to all the port critical/flush level are rejected because LCL_PTR_N−**CABPPCM.RPC** is bigger than **CABPWMLC.WMFL**	All1/All1

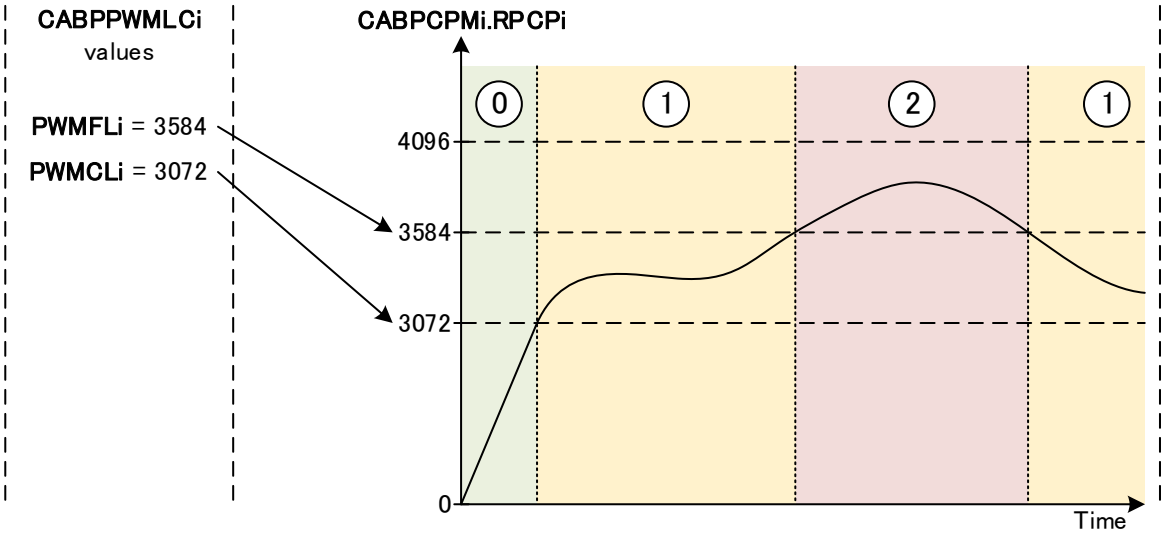
Fig 5.2: Example for Global level-based watermark setting (LCL_PTR_N == 4096)

5.1.3 Per-Port level-based watermark i (i=0..PORT_N-1)

The Per-Port level-based watermark aims at rejecting frames with targeted IPVs depending on the number of pointers are currently used by the switch (**CABPCPMi.RPCPi** register). This function is controlled by the Common Agent (through **WM.CRITICAL[i]** and **WM.FLUSH[i]**), however, the data loss is performed by the Forwarding Engine [FWD]. Note that **WM.CRITICAL** and **WM.FLUSH** interfaces can also be set by the Global level-based watermark (one of these pins will be set if the per-Port level-based watermark OR the global level-based watermark is in the required conditions to set it).

The Per-Port level-based watermark use **CABPPWMLCi.PWMFLi** and **CABPPWMLCi.PWMCLi** registers**.**

Fig 5.3 gives an example for Per-Port Level-based watermark setting for port i for and LCL_PTR_N == 4096. Note that **CABPCPMi.RPCPi** variations in Fig 5.3 are arbitrary and do not correspond to anything relevant.



	Explanation	WM.CRITICAL[i]/ WM.FLUSH[i] values
⓪	No descriptor is rejected because **CABPPWMLCi.PWMCLi** is bigger than **CABPCPMi.RPCPi**	0/0
①	Descriptors associated to all the port critical level are rejected **CABPPWMLCi.PWMCLi** is smaller than **CABPCPMi.RPCPi**	1/0
②	Descriptors associated to all the port critical/flush level are rejected **CABPPWMLCi.PWMFLi** is smaller than **CABPCPMi.RPCPi**	1/1

Fig 5.3: Example for Per-Port level-based watermark setting for port i (LCL_PTR_N == 4096)

5.2 Pause function

If a switch overflow occurs, frames in the switch will be loss at the ingress part of agents [GWCA] [RMAC]. It will result in a random loss of data independently from the actual priority of frames. To avoid so, the **“hardware pause”** function allows to selectively pause frame injection by agents [GWCA] [RMAC].

[GWCA] stops the transmission (frames R-Switch receive) from the CPU by hardware pause request.

[RMAC] controls to stop the transmission (frames R-Switch receive) from external device by sending a Pause/PFC frame by hardware pause request.

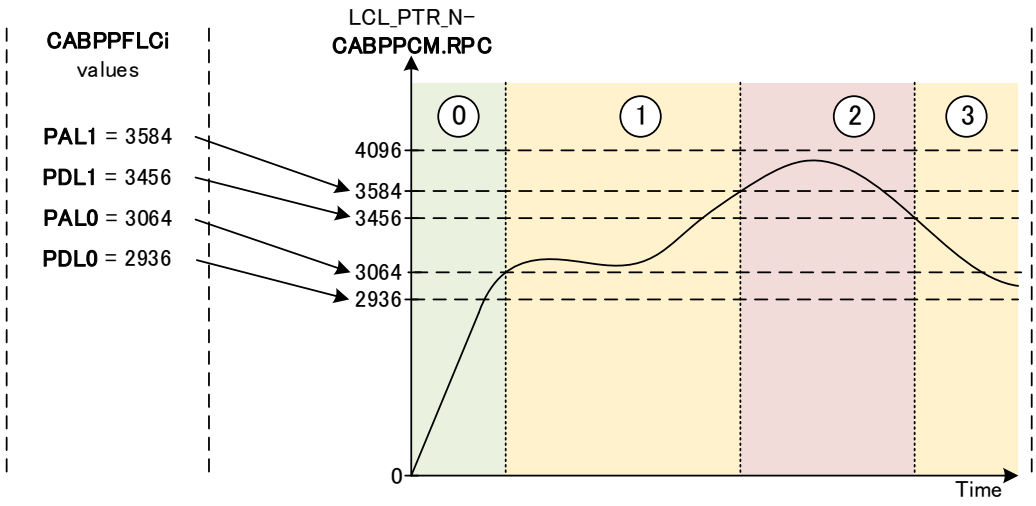
Common agent includes two pause functions, the global pause function and the per-Port pause. The pause functions aim at controlling the switch behavior in case of overflow by generating backpressure. All pause functions are running in parallel and backpressure will be applied if one or more of the pause functions decide so.

5.2.1 Global pause function

The Global pause function aims at applying backpressure on agents for targeted IPVs depending on **CABPPCM.RPC.** The pause function can pause agents with PAS_LVL_N (pause level) combinations of IPVs.

This function is controlled by the Common Agent **PAUSE[PORT_N-1:0][PAS_LVL_N-1:0]** interface, however, the back pressure is performed by the agents [TSN] [GWCA]. Note that **PAUSE** interface can also be set by the per-Port pause function (one of these pins will be set if the per-Port pause function OR the global pause function is in the required conditions to set it).

Fig 5.4 gives an example for Global pause function setting for PAS_LVL_N == 2 and LCL_PTR_N == 4096. Note that LCL_PTR_N-**CABPPCM.RPC** variations in Fig 5.4 are arbitrary and do not correspond to anything relevant.



	Explanation	PAUSE[i][0]/ PAUSE[i][1] values (for all i=1..PORT_N-1)
0	No agent paused because LCL_PTR_N-**CABPPCM.RPC** never exceeded **CABPPFLC0.PAL0** level nor **CABPPFLC1.PAL1** level	0/0
1	Agents are paused with level 0 because LCL_PTR_N-**CABPPCM.RPC** exceeded **CABPPFLC0.PAL0** level and not **CABPPFLC1.PAL1** level	1/0
2	Agents are paused with level 0 and 1 because LCL_PTR_N-**CABPPCM.RPC** exceeded **CABPPFLC0.PAL0** level and **CABPPFLC1.PAL1** level	1/1
3	Agents are paused with level 0 because LCL_PTR_N-**CABPPCM.RPC** exceeded **CABPPFLC0.PAL0** and dropped under **CABPPFLC1.PDL1**	1/0

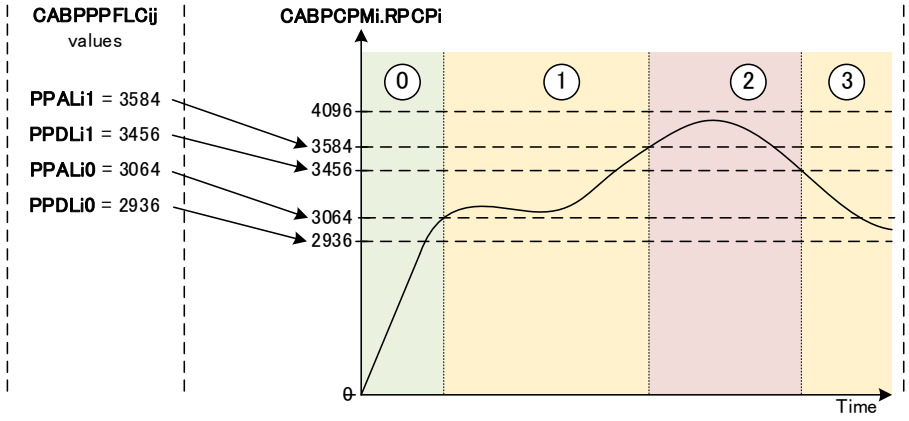
Fig 5.4: Example for Global pause function setting (PAS_LVL_N == 2) (LCL_PTR_N == 4096)

5.2.2 Per-Port Pause function i (i=0..PORT_N-1)

The Per-Port Pause function aims at applying backpressure on agents for targeted IPVs depending on **CABPCPMi.RPCPi.** The pause function can pause agents with PAS_LVL_N combinations of IPVs.

This function is controlled by the Common Agent **PAUSE[i]** interface, however, the back pressure is performed by the agents [TSN] [GWCA]. Note that **PAUSE** interface can also be set by the per-Port pause function (one of these pins will be set if the per-Port pause function OR the global pause function is in the required conditions to set it).

Fig 5.5 gives an example for Per-Port pause function setting for port i for PAS_LVL_N == 2 and LCL_PTR_N == 4096. Note that **CABPCPMi.RPCPi** variations in Fig 5.5 are arbitrary and do not correspond to anything relevant.



	Explanation	PAUSE[i][0]/ PAUSE[i][1] values
0	No agent paused because **CABPCPMi.RPCPi** never exceeded **CABPPPFLCi0.PPALi0** level nor **CABPPPFLCi1.PPALi1** level	0/0
1	Agents are paused with level 0 because **CABPCPMi.RPCPi** exceeded **CABPPPFLCi0.PPALi0** level and not **CABPPPFLCi1.PPALi1** level	1/0
2	Agents are paused with level 0 and level 1 because **CABPCPMi.RPCPi** exceeded **CABPPPFLCi0.PPALi0** level and **CABPPPFLCi1.PPALi1** level	1/1
3	Agents are paused with level 0 because **CABPCPMi.RPCPi** exceeded **CABPPPFLCi0.PPALi0** and dropped under **CABPPPFLCi1.PPDLi1**	1/0

Fig 5.5: Example for Per-Port pause function setting for port i (PAS_LVL_N == 2) (LCL_PTR_N == 4096)

5.3 Per port memory allocation function

It is possible to control the maximum and the minimum number of pointers that can be used per port thanks to **CABPULCi.MNNPNi** and **CABPULCi.MXNPNi**. Any agent will be able to use at least its minimum of pointers and at most its maximum number of pointers. In this section are given some setting examples for the memory allocation function.

5.3.1 Shared memory setting (Default setting)

In Shared memory, the Data RAM is shared by all the ports. It is the reset state of the Common Agent. Follow Fig describes the shared memory setting for LCL_PTR_N == 4096 and PORT_N == 3.

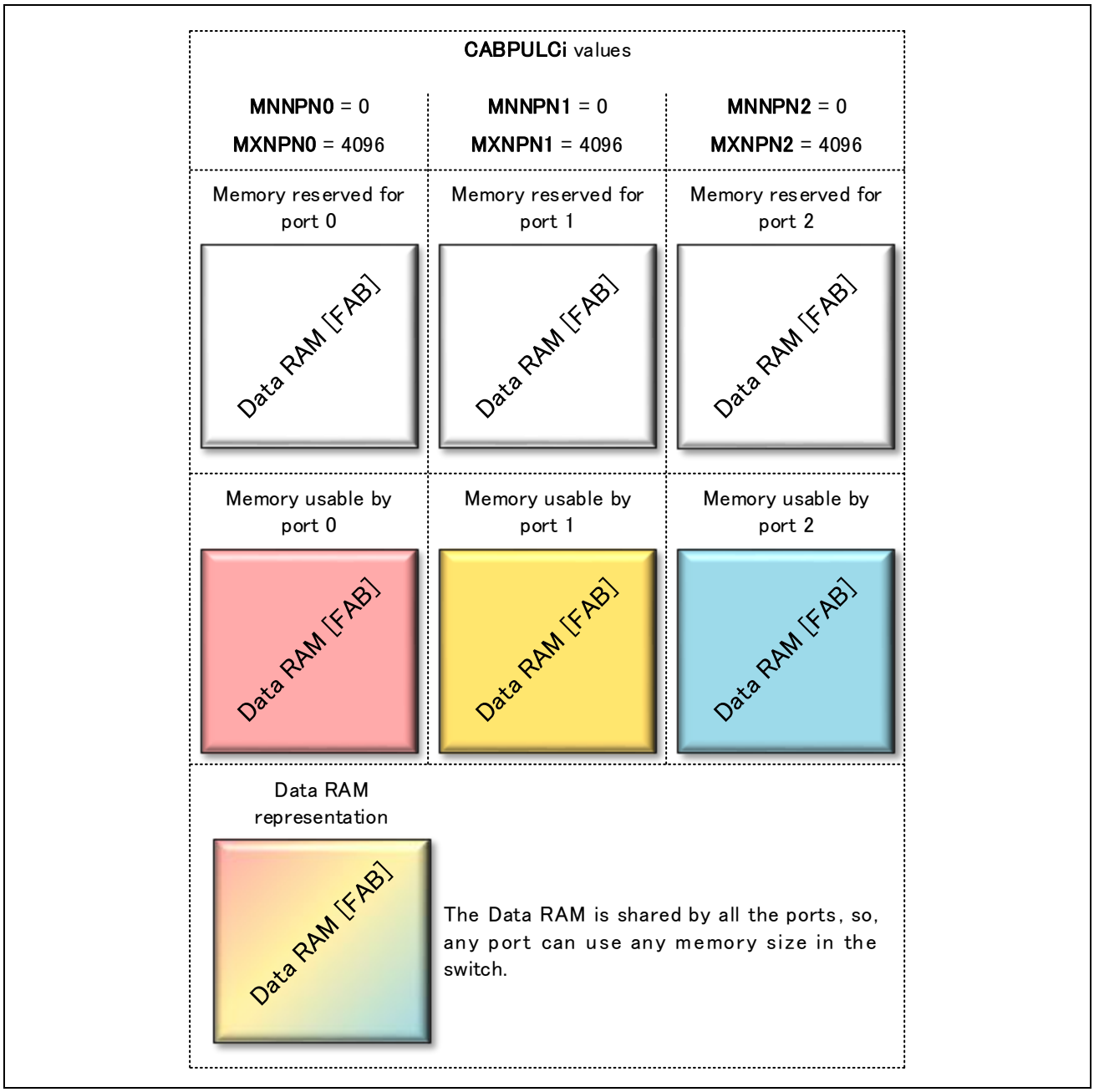


Fig 5.6: Shared memory setting (LCL_PTR_N == 4096) (PORT_N == 3)

5.3.2 Split memory setting (not recommended setting)

In Split memory, each agent has its own allocated memory. To set this mode, **CABPULCi.MXNPNi and CABPULCi.MNNPNi** should be set so their sum is equal to LCL_PTR_N.

Follow Fig describes one example of Split memory setting for LCL_PTR_N == 4096 and PORT_N == 3.

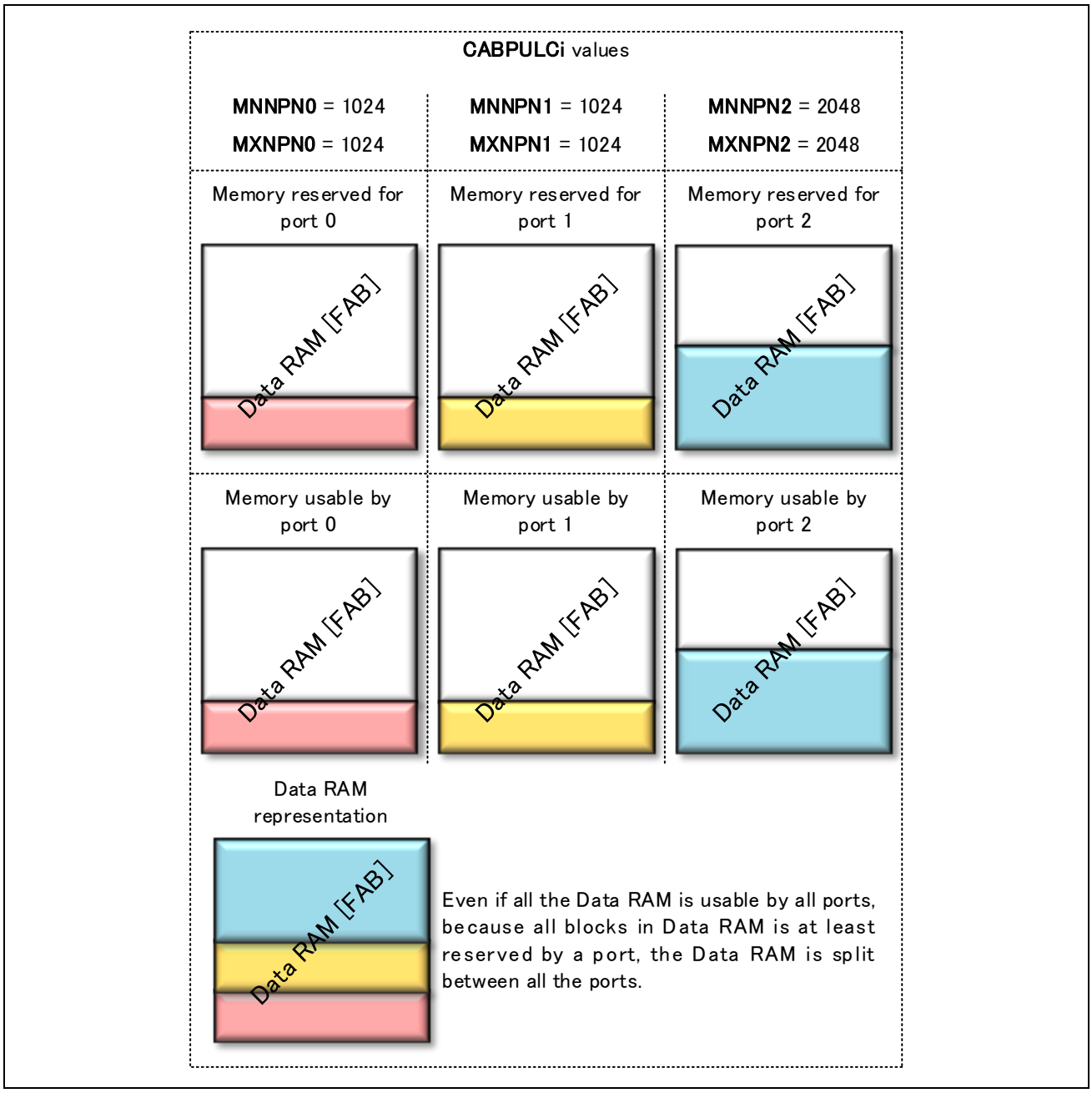


Fig 5.7: Split memory setting example (LCL_PTR_N == 4096) (PORT_N == 3)

5.3.3 Reduced memory setting 1 (Recommended setting)

In reduced memory, the data RAM is shared by all agents but some of them are not able to use all the memory because they are able to make the switch overflow (especially for CPU ports [GWCA]). To set this mode, **CABPULCi.MXNPNi** register should be used.

Follow Fig describes one example of reduced memory setting for LCL_PTR_N == 4096 and PORT_N == 3. For example, if port 2 is a fast CPU port, **CABPULCi.MXNPNi** register can limit the number of reception frames.

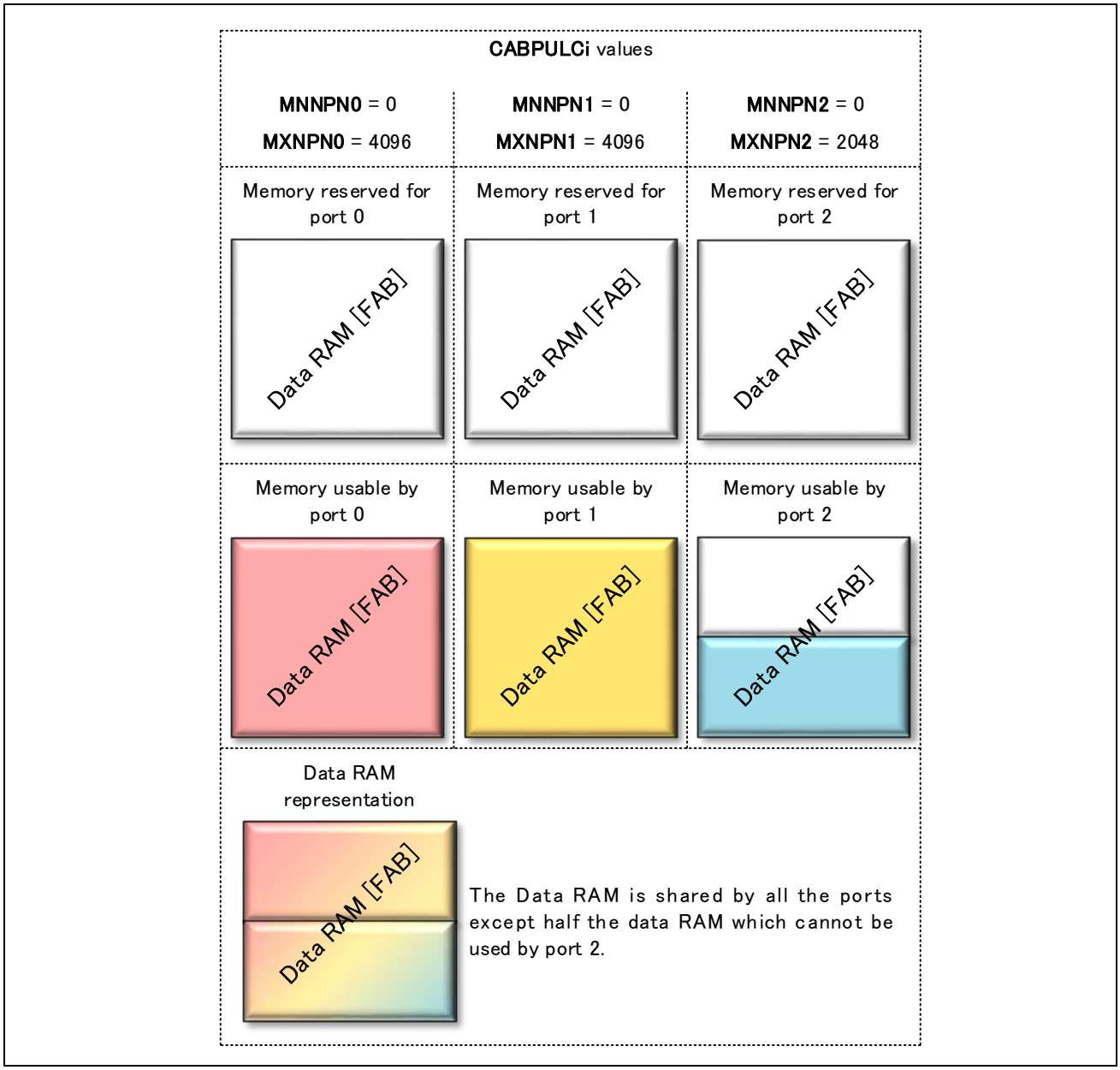


Fig 5.8: Reduced memory setting example (LCL_PTR_N == 4096) (PORT_N == 3)

5.3.4 Reduced memory setting 2 (Recommended setting)

In reduced memory, the data RAM is shared by all agents but some of them are not able to use all the memory because they are able to make the switch overflow (especially for CPU ports [GWCA]). To set this mode, **CABPULCi.MNNPNi** register should be used.

Follow Fig describes one example of reduced memory setting for LCL_PTR_N == 4096 and PORT_N == 3. For example, if port 0 and 1 are a slow Ether port, **CABPULCi.MNNPNi** register can protect the number of frame buffer.

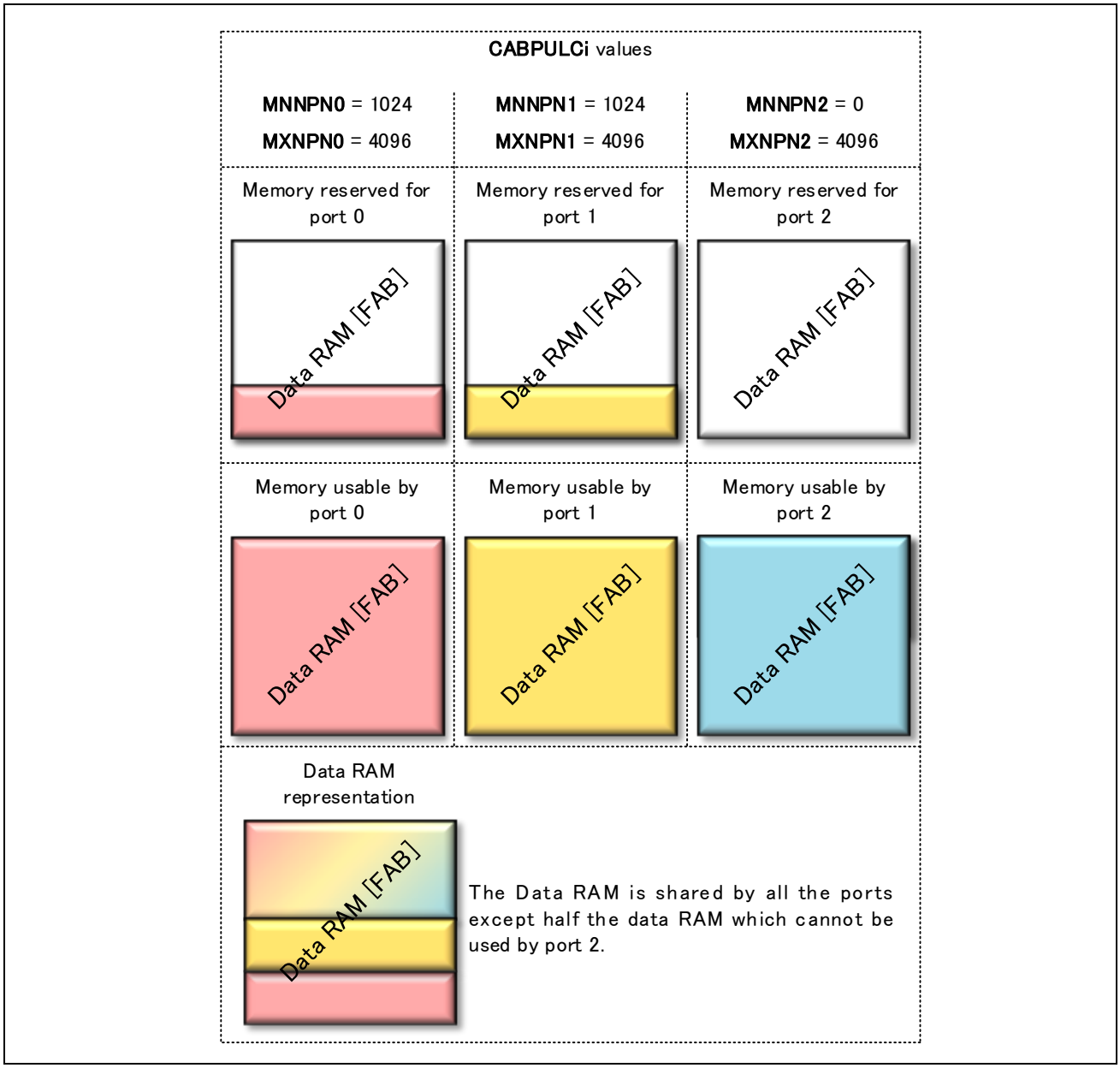


Fig 5.9: Reduced memory setting example (LCL_PTR_N == 4096) (PORT_N == 3)

5.3.5 Hybrid memory setting

All the setting described in section 5.3.1, 5.3.2 and 5.3.3 can be used together to elaborate a complex memory. It is the Hybrid memory setting.

Follow Fig describes one example of hybrid memory setting for LCL_PTR_N == 4096 and PORT_N == 4.

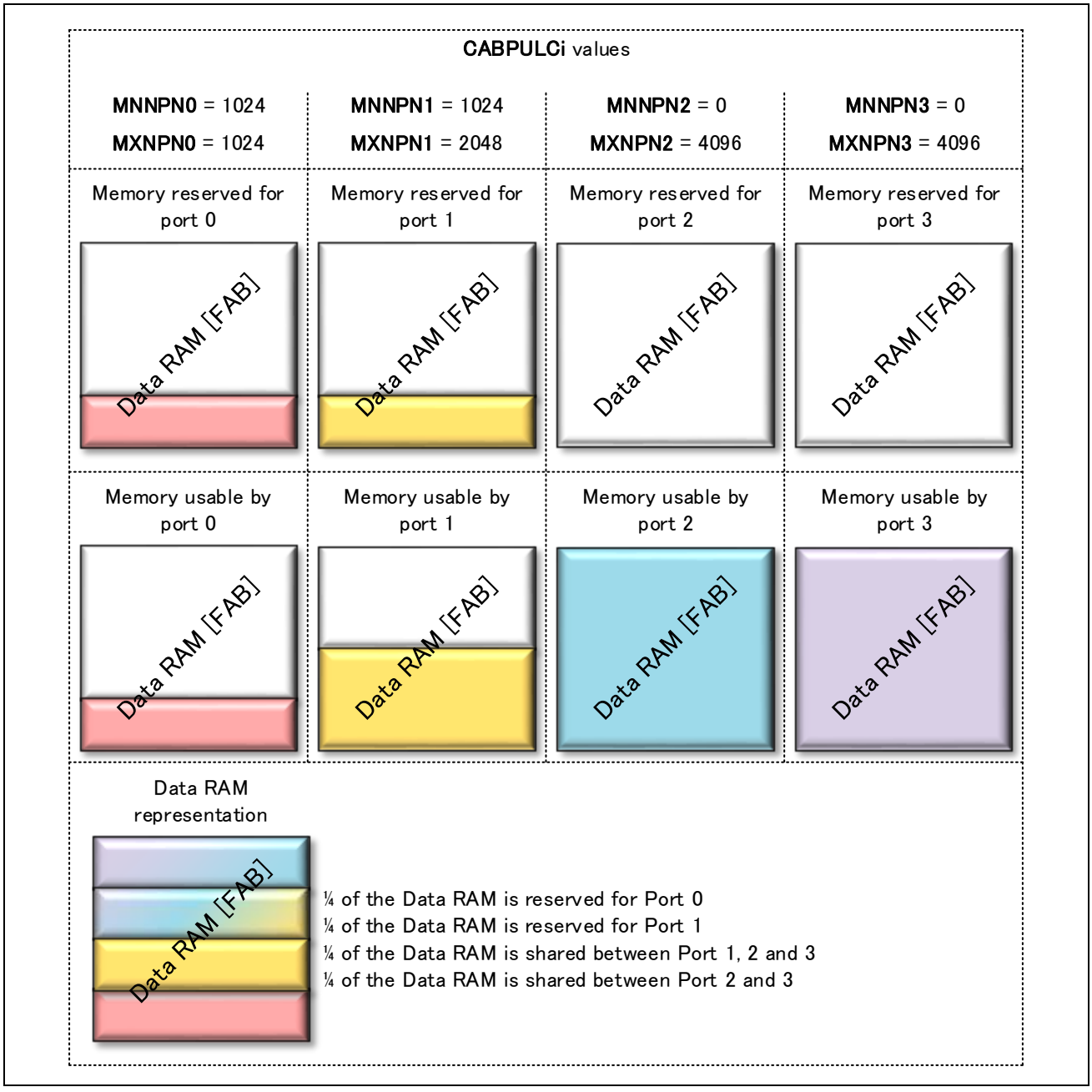


Fig 5.10: Hybrid memory setting example (LCL_PTR_N == 4096) (PORT_N == 3)

5.4 Inter-CPU interrupt function

The inter-CPU interrupt function can be used to create interrupts between CPUs and is controlled by **RSSIS** register. **RSSIS.SNSSISi** is used to create an interrupt from Secure to Non-Secure CPU and **RSSIS.NSSSISi** is used to create an interrupt from Non-Secure to Secure CPU.

This register and interrupt can notify the Non-Secure CPU that the Secure CPU has written to "Security registers". This register and interrupt can notify the Secure CPU that the Non-Secure CPU has written to "Some specific configuration registers".

The Secure CPU don’t access “permitted registers by Security registers from the Secure CPU to the Non-Secure CPU”.

6. Precautions

6.1 Precautions

NA.

6.2 Restrictions (Including known problems)

NA.

Published by Renesas Electronics Corporation