User's Manual

MFAB
(R-Switch-3 GateWay Fabric)

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

Fabric is a bus module developed by Renesas System Design. Fabric is the Renesas Switch [TOP] internal bus used to exchange data between TSN-Agent[TSN], Common Agent[COMA], GWCA[GWCA], Local RAM, TAG RAM, pointer RAM and Forwarding engine[FWD].

1.1 Features

Fabric Features are described Table 1-1.

Table 1-1 Fabric Feature List

Function		Details
Overall function	Data transfer	・Function to transfer data and access to RAMs
Data provision		・Data for Fabric ・Initial frame data + Intermediate data + Last frame data + Frame Descriptor ・Data width：LCL_RAM_DW + 48 + LCL_PTR_W bits
Control function	Time arbiter	・Arbitration between Time-critical agents [TSN] for data bus. Arbitration scheme depends on each Time critical agents [TSN] PHY link/Internal speed
	Arbitration	・LRU for Time critical-agents [TSN], Slow agents [GWCA] and Common Agent [COMA] error read requests arbitration. ・LRU between Slow agents [GWCA] for read/write bus arbitration. ・Strict priority like arbitration between time arbiter, Error LRU, and slow LRU. Priority: time arbiter > slow LRU > Error LRU
	RAM access	・Function that writes and reads to RAMs
	Transfer write request to Forwarding Engine	・Function to transfer write request sent from TSN Agents [TSN] or GWCAs [GWCA] to Forwarding Engine
	ECC error detection	・Function to correct 1 bit ECC errors and detect 2 bit ECC errors

1.2 Fabric block diagram

Fig 1.1 shows Fabric block diagram.

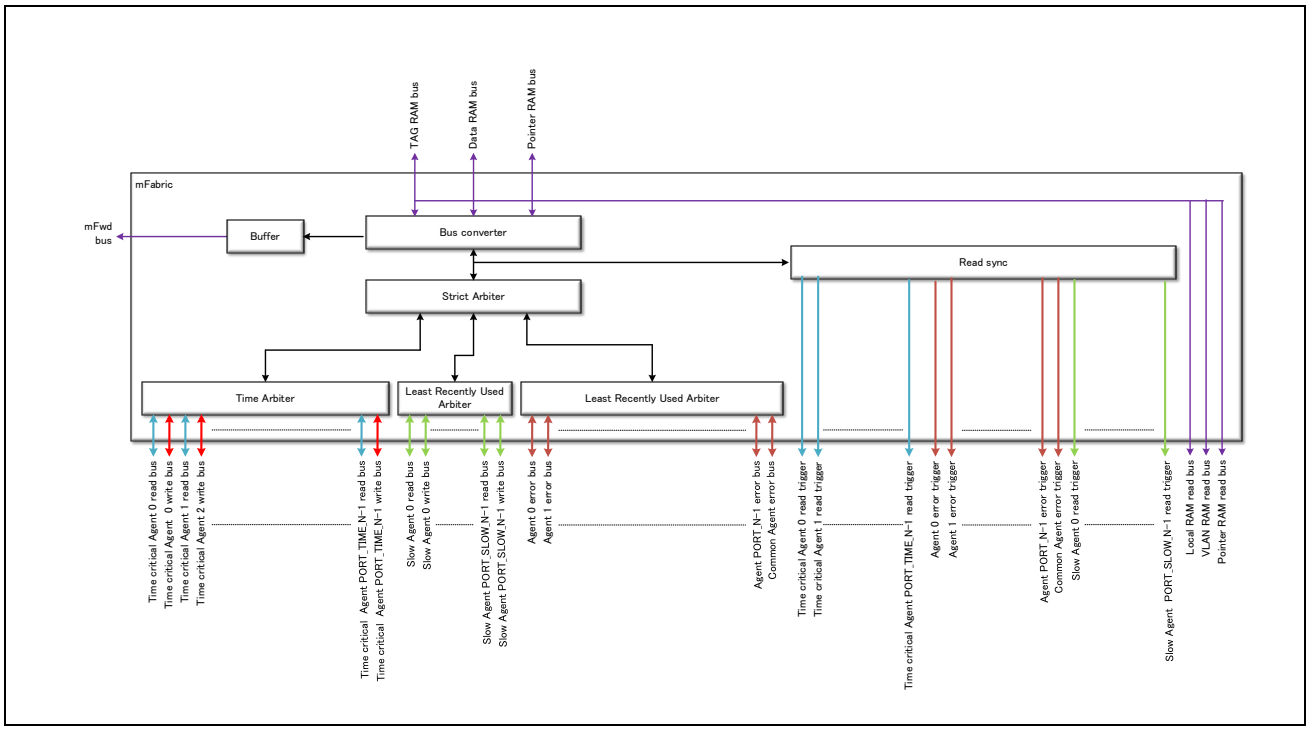

Fig 1.1: Fabric Block diagram

Table 1-2 Fabric Functional Blocks

Block name	Function
Time arbiter	Function to share the fabric throughput per agent depending on agent modes and PHY speed.
Least Recently Used arbiter	Arbiter between several buses. (LRU) If multiple error read requests collide, LRU arbitration is executed on a cycle by cycle basis.
Strict arbiter	Arbiter between several buses. If requests from Time arbiter, Round robin arbiter and GWCA collide, the higher priority request will be processed first. Priority ordering: Time-critical Write/Read request > Slow Write/Read request > Agent Error Read request
Read sync	Module that generates a signal indicating whether data read from RAM is valid or invalid.
Bus converter	Module that converts the internal access request in RAM protocol
Buffer	Register slices to meet design timings.

2. Parameter list

Fabric global parameter list is shown in Table **2-1**.

Fabric local parameter list is shown in Table **2-2**.

Table 2-1 Global parameter list

Parameter Name	**RSW3.0 Values**	**Explanation**
Port Number		
PORT_TSNA_N	13	TSN Agent Number [TSN]
PORT_GWCA_N	2	CPU Agent Number [GWCA]
Local RAM		
LCL_RAM_SZ	1024	Local RAM size in Kbytes (only for data RAM part)
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) (only for data RAM part)
LCL_RAM_N	2	Number of Local RAM The case of LCL_RAM_DW=128, set this parameter to 1. In other cases, set this parameter to 2.

Table 2-2 Local parameters

Parameter Name	RSW3.0 Values	Explanation
Port Number		
PORT_N	15	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
PORT_TIME_N	13	Number of time-critical ports on the switch (plugged to the Fabric time arbiter)
PORT_TIME_W	4	Number of time-critical ports on the switch bus width
PORT_SLOW_N	3	Number of non-time-critical ports on the switch (plugged to the Fabric LRU arbiter)
PORT_SLOW_2W	3	Number of non-time-critical ports on the switch *2 bus width
Local RAM		
LCL_PTR_N	8192	Pointer number to address local RAM
LCL_PTR_W	13	Local RAM Pointer width
LCL_RAM_AW	14	Local RAM address width
LCL_RAM_DW	512	Local RAM data width
LCL_DATA_SIZE	6	Internal Bus size width

3. Register

3.1 Register list

The Fabric register list is described in Table **3-1**. MFRO (Fabric Register Offset) indicates base address of address space allocated to Fabric by the system.

Access Mode:

- Any: Register can be accessed in any mode.

Table 3-1: List of Fabric registers

Offset/Address	Register name	Abbreviation
	No register in fabric	

Restrictions:

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

3.2 Register detailed explanation

This section describes SFR details.

4. Register utilization

4.1 Software flows

No SFRs in mFabric.

5. Functional details

5.1 Data representation in RAMs

Agents connected to Fabric will use TAG, Pointer and Data RAM to store their data. Because of peripheral IPs such as Forwarding Engine, the data saving/fetching patterns described in this section should be respected by Agents.

5.1.1 RAM address dependencies

When an agent access to fabric, it can decide to access to one, two or all the RAMs in one access (thanks to *_den_*, *_pen_* and *_ten_* signals). However, there is only one address per one access. The address at which RAMs will be accessed will be calculated from the address given to the Fabric. The address calculation for Data, TAG and Pointer RAMs are respectively described in equations (1), (2) and (3). Fig. 5.1 describes the first addresses mapping to RAMs for LCL_RAM_BSZ = 128.

(1) *dataAddress = fabricAddress*

(2) *tagAddress = fabricAddress[LCL_RAM_AW-1:LCL_RAM_AW-LCL_PTR_W]*

(3) *pointerAddress = fabricAddress[LCL_RAM_AW-1:LCL_RAM_AW-LCL_PTR_W]*

Where:

- *fabricAddress* is the address given by any agent to Fabric.

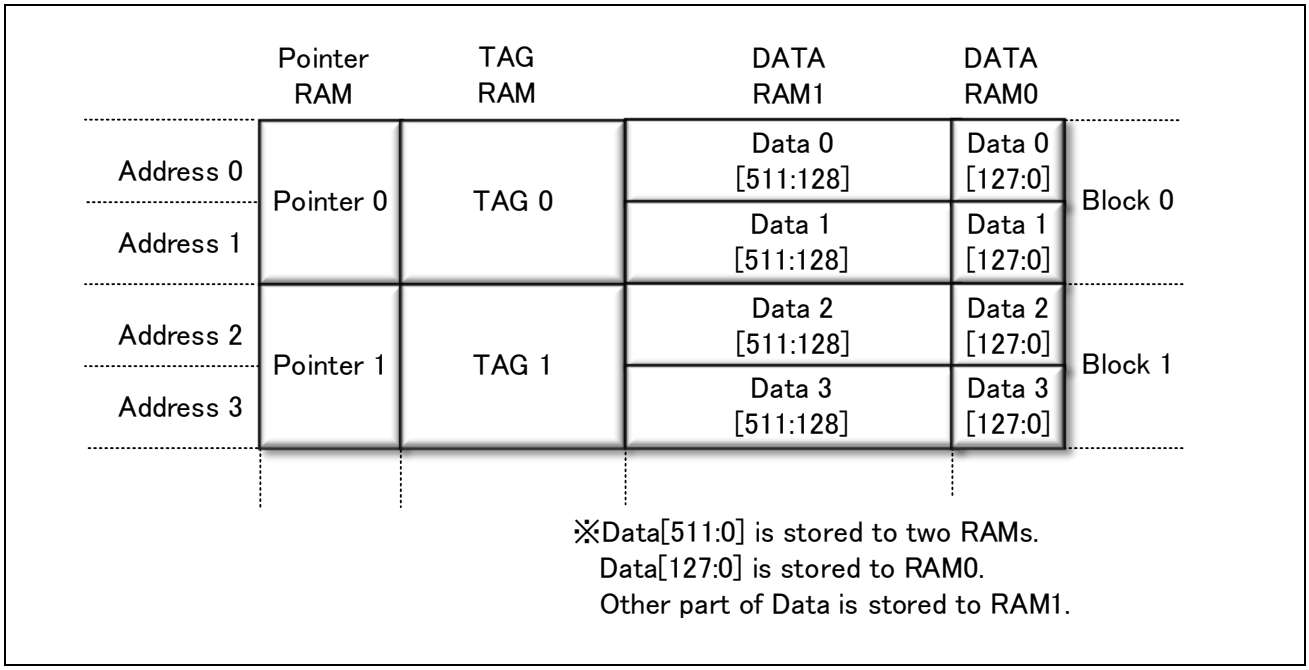

Fig. 5.1: RAM first address representation for LCL_RAM_BSZ = 128 and LCL_RAM_DW=512

5.1.2 RAM access by agents

While written in RAMs, the Agents should respect a given structure defined by the following rules and described in Fig. 5.2, Fig. 5.3 and Fig. 5.4.

Restrictions:

- HW: A frame data should be written in a block in order from low address to high address.
- HW: When a frame is contained in several blocks, the next block pointer should be written in the previous block pointer RAM to create a block chain.
- HW: In any block containing the first part of a frame, the bits[127:0] of the beat 0 address is used for the descriptor.
- HW: While writing in a frame first block, the TAGs should be written with the first beat of data.
- HW: A pointer should always been written/read while accessing a block for the first time if needed.
- HW: The descriptor should always be written last and read first.
- HW: Type signals should be constant inside a frame. Value 0 is used for E-frames and 1 for P-frames.
- HW: Size signal can only be used for the last beat of data.
- HW: Cmd signals is used to signals what is being written to the RAMs. Value 0 is used for start-of-frame, Value 2 for end-of-frame and Value 3 for descriptor. All other accesses are done with value 1.
  When the flame length is “LCL_RAM_DW-128bit” or less, only value 0 and value 3 are used for Cmd.

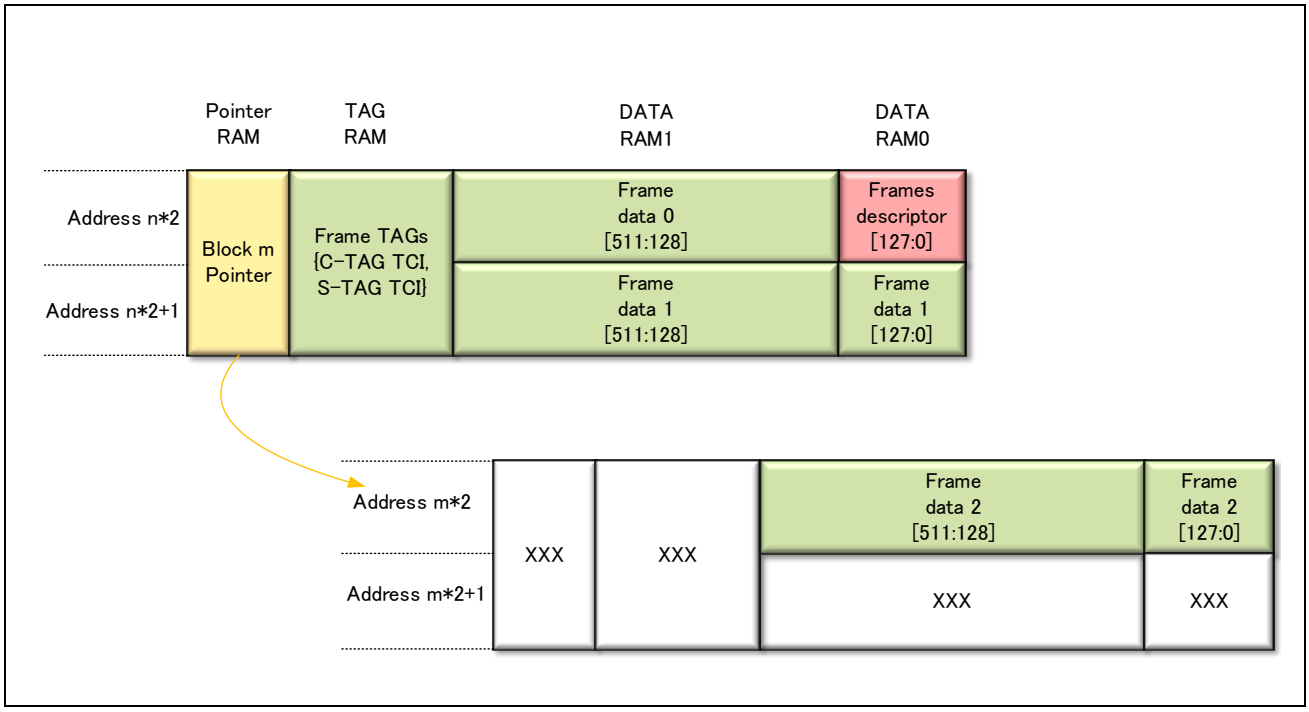

Fig. 5.2: Fabric data structure for LCL_RAM_BSZ = 128 and LCL_RAM_DW=512

Notes:

- Only S-TAGs and C-TAGs will be saved in the TAG RAM. The R-TAG will be taken by Forwarding Engine [FWD] and saved in the agent descriptor RAM [GWCA][TSN].

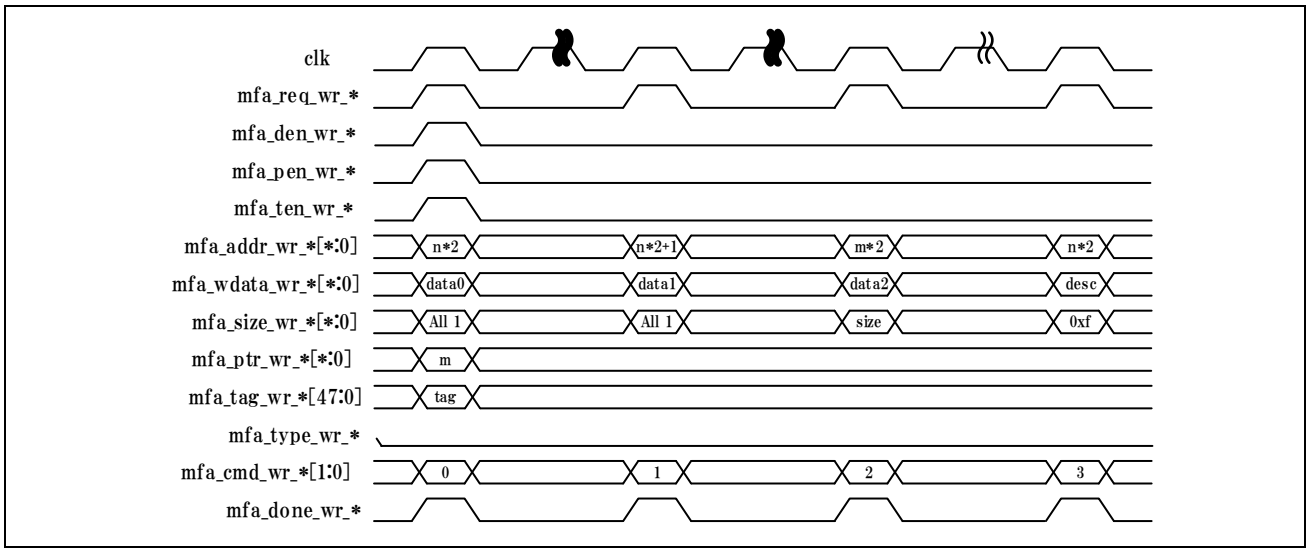

Fig. 5.3 :Frame corresponding fabric write accesses

When LCL_RAM_DW is 256 or more, data0 and descriptor are stored at same RAM address. In this case, mfab overwrites only descriptor part (128bit) at timing of cmd=3.

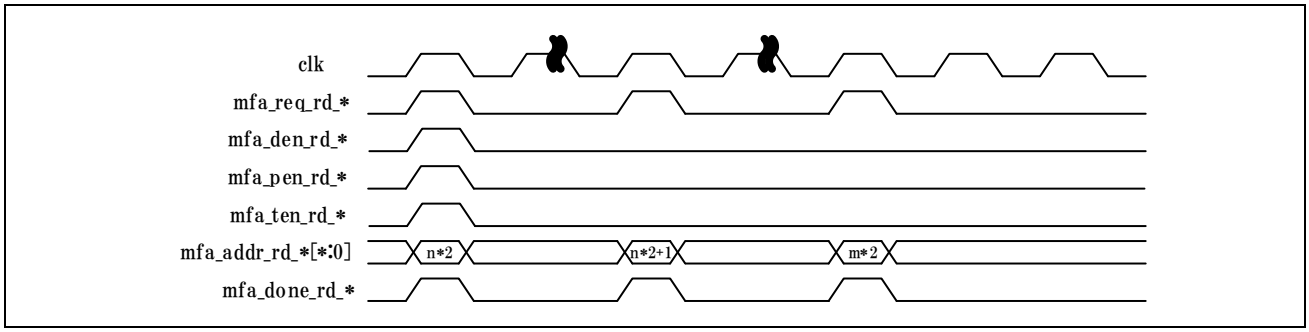

Fig. 5.4: Frame corresponding fabric read accesses

5.2 Fabric Time arbiter

Time arbiter controls done signal of TSN Agent [TSN] Write and Read busses depending on link speed. The faster a link is the more the corresponding agent will be able access to the local RAM. Time arbiter supports PHY link speeds from 10Mbps to 10Gbps.

- Values is : 10Mbps, 100Mbps, 1Gbps, 2.5Gbps, 5Gbps, 10Gbps

5.2.1 Agents done signal distribution

Table 5-1 shows done signal distribution depending on PHY link speed. Any disabled agent has a *doneSignalDistributionValue* equal to 0.

Table 5-1 Done signal output ratio by combination of PHY link speed

PHY link speeds of time-critical agents in OPERATION mode [TSN]	***doneSignalDistributionValue[cycle]***
Only one Agent enabled	1
PHY link speed of all Agents is the same	1
The case of mixing PHY link speed	The vaule will be a simple integer ratio
[example1] 100Mbps : 1Gbps	1 : 10
[example2] 1Gbps : 2.5Gbps	2 : 5 (refer to Fig. 8.5)
[example3] 100Mbps : 100Mbps : 1Gbps	1 : 1 : 10
[example4] 1Gbps : 1Gbps : 2.5Gbps : 10Gbps	2 : 2 : 5 : 20
[example5] 10Mbps : 100Mbps : 1Gbps : 2.5Gbps : 5Gbps : 10Gbps	1 : 10 : 100 : 250 : 500 : 1000

5.2.2 Done signal patterns

Fig. 5.5 shows a done signal pattern example where time-critical port 0 has a 1Gbps link and time-critical port 1 has a 2.5Gbps link. When a request signal is not asserted at the timing when the corresponding done signal should have been asserted the done signal is not asserted and the handshake doesn't happen. In Fig. 5.5 the request signals are always asserted to make the done pattern visible but, in a real system, request signals can be asserted anytime.

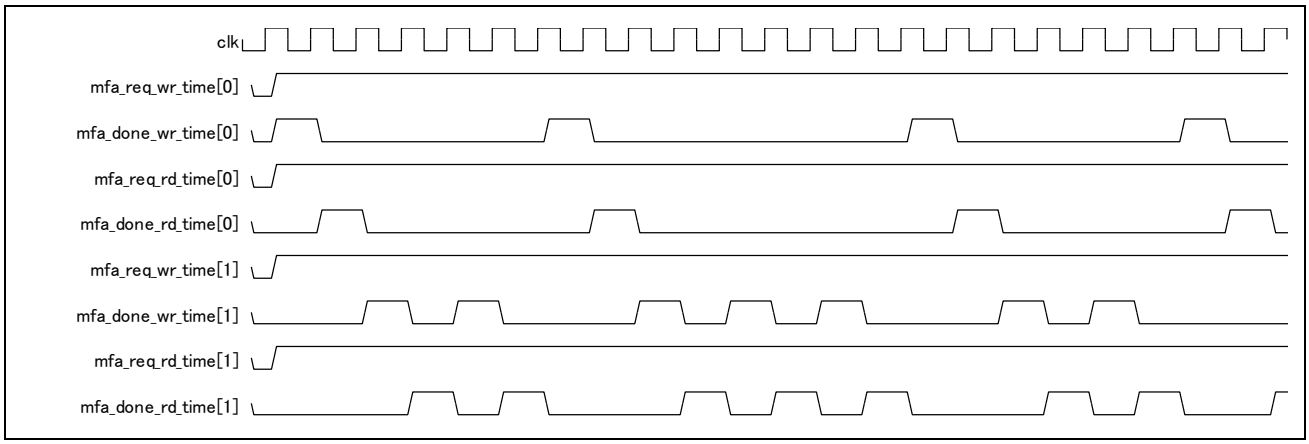

Fig. 5.5: Done signal pattern example (TSNA0 at 1Gbps and TSNA1 at 2.5Gbps)

The done signal pattern has inherent properties that are always respected:

- For a static configuration the done pattern is cyclic with a constant cycle time *doneCycleClk* described in equation (4).
- During a period of *doneCycleClk* clocks starting from anywhere, a time-critical agent receives for both read and write buses a number of done signals equal to its associated done signal distribution value (Table **5-1**).

(4) $doneCycleClk[cycle] = (\Sigma_{PORT\_TIME\_N}\ doneSignalDistributionValue[cycle])*2$

Where:

- $\Sigma_{PORT\_TIME\_N}$ *doneSignalDistributionValue* is the sum of all time-critical agents in OPERATION mode *doneSignalDistributionValue* (Table **5-1**).

Restriction:

- HW: Done signals are only given to ethernet agents in OPERATION mode [TSN].
- HW: If an unknown link speed is set for time-critical port i [TSN]
- (mfa_xmii_speed[4*(i+1)-1:4*i] > 4'b1011 [RMAC]), Fabric will consider port i as disabled and port i will not be able to access the time arbiter.

Note:

If the PHY link speed is changed, there is a period when Fabric doesn't output the done siganl.

5.2.3 Fabric jitter

When a TSN Agent reads data using fabric time arbiter, some jitter is induced in the reading process by Fabric and it should be taken in account while using TAS [802.1Qbv]. The fabric jitter can be calculated for an agent using equation (6).

(6) *fabricJitter[ns]* = (⌈ *doneCycleClk[cycle]* / *doneSignalDistributionValue[cycle]* ⌉ + (*PortTimeOperation[number]*-1)*2) * *clk_period[ns]*

Where:

- *doneCycleClk* is detailed in section 5.2.2.
- *doneSignalDistributionValue* is corresponding agent value detailed in section 5.2.1.
- *PortTimeOperation* is the number of TSN Agents in OPERATION mode.
- *clk_period* is the clock clk period.

Restrictions:

- SW: The *fabricJitter* equation takes in account the number of TSN Agents [TSNA] in OPERATION mode and also *doneCycleClk*. In fact, enabling/disabling an agent or changing an agent PHY speed will indeed affect the fabric jitter value. However, reconfiguration TAS [802.1Qbv] jitter in OPERATION is not a recommended option so, the maximum number of TSN Agents in OPERATION mode and their corresponding maximum PHY speed should be determined to calculate the worst fabricJitter per agent and it should be kept to the same value all the time.

5.2.4 Fabric minimum latency

When a TSN Agent reads data using fabric time arbiter, some latency is induced in the reading process by Fabric and it should be taken in account while using TAS [802.1Qbv]. The fabric minimum latency can be calculated for an agent using equation (7).

(6) *fabricLatency[ns]* = (⌊ *doneCycleClk[cycle]* / *doneSignalDistributionValue[cycle]* ⌋ - 1) * *clk_period[ns]*

Where:

- *doneCycleClk* is detailed in section 5.2.2.
- *doneSignalDistributionValue* is corresponding agent value detailed in section 5.2.1.
- *clk_period* is the clock clk period.

Restrictions:

- SW: The *fabricLatency* equation takes in account *doneCycleClk*. In fact, enabling/disabling an agent or changing an agent PHY speed will indeed affect the fabric latency value. However, reconfiguration TAS [802.1Qbv] latency in OPERATION is not a recommended option so, if ethernet agent are disabled or PHY speed is changed during OPERATION, it is recommended to set the *fabricLatency* to 2**clk_period[ns]* and add the *fabricLatency* calculated value in the worst case to *fabricJitter*.
- SW: When preemption is enabled for a port, *fabricLatency* should be set to 0ns and *fabricLatency* calculated value when all agents are enabled should be added to *fabricJitter*.

5.3 Fabric arbitration

If multiple requests collide, request is determined by arbitration.

(1) Round robin arbitration (LRU)

When multiple error read requests collide, the priority of the request is determined by the round robin arbitration. Fabric selects the least recently used error read request. Round robin arbitration at Fabric is LRU. The error read requests that accessed the longest ago take precedence.
The round robin arbitration is executed on a cycle by cycle basis.

(2) Strict arbitration

If requests from Time arbiter and Round robin arbiter of error and GWCA collide, the priority of the request is determined by strict arbitration.

Priority ordering: Time-critical Write/Read request > Slow Write/Read request > Agent Error Read request

5.4 Source port generation (Bit i of [POPRT_N-1:0] assignment)

Fabric gives automatically a source port number to each Agent depending on the bus they are accessing to. Table **5-2** shows the relation between the Bus accessed by an agent and its source port number.

Table 5-2: Source port number assignment

Bus from which agent access the fabric	**Source port number (Number of i) [i] of [PORT_N-1:0]**
TSNA0	0
TSNA1	1
TSNA2	2
TSNA3	3
TSNA4	4
TSNA5	5
TSNA6	6
TSNA7	7
TSNA8	8
TSNA9	9
TSNA10	10
TSNA11	11
TSNA12	12
GWCA0	13
GWCA1	14

6. Precautions

6.1 Precautions

NA.

Published by Renesas Electronics Corporation

ルネサス エレクトロニクス株式会社