28.5 Advanced Routing Unit (ARU)

28.5.1 Overview

The Advanced Routing Unit (ARU) is a flexible infrastructure component for transferring 53 bit wide data (five control bits and two 24 bit values) between several sub-modules of the GTM core in a configurable manner.

Since the concept of the ARU has already been described in the paragraph “ARU routing concept”, this section only describes additional ARU features that can be used by the software for configuring and debugging ARU related data streams.Also the definition of 'streams' and 'channels' in the ARU context is done in “ARU routing concept”.

The principle of ARU data routing is described in “ARU Write Address Overview”. In the real GTM implementation the ARU serves in parallel per clock period two individual data destinations, one destination at port ARU-0 and at port ARU-1. Both ARU ports ARU-0 and ARU-1 are running by default in parallel but can be configured in dynamic routing mode (see below) to run in an individual mode.

As already defined in the “ARU routing concept”, the ARU read ID is the address of the data source that is configured in the data destination module. These ARU read ID's are selected by the individual counter of ARU ports ARU-0 and ARU-1.

Via the ARU ports ARU-0 and ARU-1 with each ARU read ID two independent GTM sub-modules are addressed and served. The combination of ARU port (ARU-0 or ARU-1) and the ARU read ID addresses one ARU wdata source (i.e. the ARU write port of a GTM sub-module).

The assignment of ARU write ports of GTM sub-modules to the ARU ports ARU-0 and ARU-1 and the ARU read ID's is device depending and can be found in the corresponding sub-chapter.

28.5.2 Special Data Sources

Besides the addresses of the sub-module related data sources as described in product specific appendix, the ARU provides two special data sources that can be used for the configuration of data streams. These data sources are defined as follows:

Address 0x1FF: Data source that provides always a 53 bit data word with zeros. A read access to this memory location will never block a requesting data destination.

Address 0x1FE: Data source that never provides a data word. A read access to this memory location will always block a requesting data destination. This is the reset value of the read registers inside the data destinations.

Address 0x000: This address is reserved and can be used to bring data through the ARU registers **ARU_DATA_H** and **ARU_DATA_L** into the system by writing the write address 0x000 into the **ARU_ACCESS** register. This means that software test data can be brought into the GTM by the CPU.

28.5.3 ARU Access via AEI

Besides the data transfer between the connected sub-modules, there are two possibilities to access ARU data via the AEI.

28.5.3.1 Default ARU Access

The default ARU access incorporates the registers **ARU_ACCESS**, which is used for initiation of a read or write request and the registers **ARU_DATA_H** and **ARU_DATA_L** that provide the ARU data word to be transferred.

The status of a read or write transfer can be determined by polling specific bits in register **ARU_ACCESS**. Furthermore the *acc_ack* bit in the interrupt notify register is set after the read or write access is performed to avoid data loss e.g. on access cancelation.

A pending read or write request may also be canceled by clearing the associated bit.

In the case of a read request, the AEI access behaves as a read request initiated by a data destination of a module. The read request is served by the ARU immediately when no other destination has a pending read request. This means, that an AEI read access does not take part in the scheduling of the destination channels and that the time between two consecutive read accesses is not limited by the round trip time.

On the other hand, the AEI access has the lowest priority behind the ARU scheduler that serves the destination channels. Thus, in worst case, the read request is served after one round trip of the ARU, when all destination channels would request data at the same point in time.

In the case of the write request, the ARU provides the write data at the address defined by the ADDR bit field inside the **ARU_ACCESS** register.

To avoid data loss, the reserved ARU address 0x0 has to be used to bring data into the system. Otherwise, in case the address specified inside the ADDR bit field is defined for another sub-module that acts as a source at the ARU data loss may occur and no deterministic behavior is guaranteed.

This is because the regular source sub-module is not aware that its address is used by the ARU itself to provide data to a destination.

It is guaranteed that the ARU write data is send to the destination in case of both modules want to provide data at the same time.

Configuring both read and write request bits results in a read request, if the write request bit inside the register isn't already set. The read request bit will be set but not the write request bit. The following table describes the important cases of the bit 12 (RREQ) and bit 13 (WREQ) of the **ARU_ACCESS** register:

Table 12 WREQ and RREQ in ARU_ACCESS register

AEI write access: aei_wdata (13:12)	actual value of ARU_ACCESS(13:12)	next value of ARU_ACCESS(13:12)	comment
0 0	0 1	0 0	cancel read request
0 0	1 0	0 0	cancel write request
0 1	1 0	1 0	unchanged register
1 0	0 1	0 1	unchanged register
1 1	0 0	0 1	both read and write request results in a read request
1 1	1 0	1 0	as before but WREQ bit is already set -> unchanged register

28.5.3.2 Debug Access

The debug access mode enables to inspect routed data of configured data streams during runtime.

The ARU provides two independent debug channels, whereas each is configured by a dedicated ARU read address in register **ARU_DBG_ACCESS0** and **ARU_DBG_ACCESS1** respectively.

The registers **ARU_DBG_DATA0_H** and **ARU_DBG_DATA0_L** (**ARU_DBG_DATA1_H** and **ARU_DBG_DATA1_L**) provide read access to the latest data word that the corresponding data source sent through the ARU.

Any time when data is transferred through the ARU from a data source to the destination requesting the data the interrupt signal *ARU_NEW_DATA0*_IRQ (*ARU_NEW_DATA1*_IRQ) is raised.

For advanced debugging purposes, the interrupt signal can also be triggered by software using the register **ARU_IRQ_FORCINT**.

Please note, that the debug mechanism should not be used by the application, when a HW-Debugger is used to trace the ARU communication. In that case, the debug registers are used by the HW-Debugger to specify the ARU streams that should be traced.

28.5.4 ARU dynamic routing

A dynamic routing feature of the ARU is implemented and can be configured using the additional AEI registers**:**,

- **ARU_CTRL**
- **ARU_[x]_DYN_CTRL**
- **ARU_[x]_DYN_RDADDR**
- **ARU_[x]_DYN_ROUTE_LOW**
- **ARU_[x]_DYN_ROUTE_HIGH**
- **ARU_[x]_DYN_ROUTE_SR_LOW**
- **ARU_[x]_DYN_ROUTE_SR_HIGH**

For further information see the register part of this chapter.

28.5.4.1 Dynamic routing - CPU controlled

The dynamic routing feature can be enabled separately for ARU-0 and ARU-1 by setting the corresponding bit fields of the register **ARU_CTRL**.

The enabling of the dynamic routing feature is synchronized to the normal routing scheme if ARU master ID-0 is addressed. The dynamic route will started with additional ARU master DYN_READ_ID0.

With the dynamic routing feature it is possible to insert additional ARU master ID's, DYN_READ_IDy (y:0-5), in a defined manner into the normal ARU routing scheme.

While inserting additional ARU master ID's the normal ARU routing scheme is paused. Therefore please consider that inserting additional ARU master ID's will lengthen the normal routing scheme.

It is possible to configure 6 additional ARU master ID's in the **ARU_[x]_DYN_ROUTE_LOW/_HIGH** registers for both ARU-0 and ARU-1.

In the bit field **DYN_CLK_WAIT** of **ARU_[x]_DYN_ROUTE_HIGH** register the number of clock cycles has to be configured, after which one of the additional ARU master ID's will be inserted.

After each configured number of clock cycles the defined ARU master ID's will be inserted cyclic one after each other in the following manner:
… -> DYN_READ_ID0 -> DYN_READ_ID1 -> DYN_READ_ID2 -> DYN_READ_ID3 -> DYN_READ_ID4 -> DYN_READ_ID5 -> DYN_READ_ID0 -> ….

In the shadow registers **ARU_[x]_DYN_ROUTE_SR_LOW/_HIGH** further 6 ARU master, DYN_READ_IDy (y:6-11), can be configured.

The bit DYN_UPDATE_EN in the **ARU_[x]_DYN_ROUTE_SR_HIGH** register controls whether the **ARU_[x]_DYN_ROUTE_LOW/_HIGH** registers are updated from its shadow registers except DYN_UPDATE_EN, it is not updated. The update is executed once after writing **ARU_[x]_DYN_ROUTE_SR_HIGH**. If update started DYN_UPDATE_EN is reset.

With the DYN_ROUTE_SWAP option in the **ARU_[x]_DYN_CTRL** register it is possible to swap the registers **ARU_[x]_DYN_ROUTE_LOW/HIGH** with its shadow registers **ARU_[x]_DYN_ROUTE_SR_LOW/HIGH**. The swapping is executed always after the 6 ARU master DYN_READ_ID's are inserted. So it is possible to insert a maximum of 12 ARU master DYN_READ_ID's cyclic after each configured number of clock cycles. If swap started DYN_UPDATE_EN is reset.

Setting the bit field **DYN_CLK_WAIT** of **ARU_[x]_DYN_ROUTE_HIGH** register to zero, only the defined ARU master DYN_READ_ID's will be executed. The normal ARU routing scheme is stopped.

Setting the bit field **DYN_CLK_WAIT** of **ARU_[x]_DYN_ROUTE_HIGH** register to 15, only the normal ARU routing scheme is executed. Inserting of additional ID's is stopped.

To reset the ARU caddr counter and ARU dynamic route counter set bit ARU_ADDR_RSTGLB of **CMU_GLB_CTRL** following by a write access to register **CMU_CLK_EN**.

28.5.4.1.1 Dynamic routing ring mode

In dynamic routing ring mode it is possible to use all 24 DYN_READ_ID's from both ARU-0 and ARU-1 by setting bit field ARU_DYN_RING_MODE in **ARU_CTRL** register to 1. In this mode all 4 registers **ARU_[x]_DYN_ROUTE_LOW/_HIGH** and **ARU_[x]_DYN_ROUTE_SR_LOW/_HIGH** are connected as a ring, so all 24 DYN_READ_ID's can be used from both ARU's. The ring structure is shown in **Figure 24**. The data register shift direction is shown by the arrows in the ring.

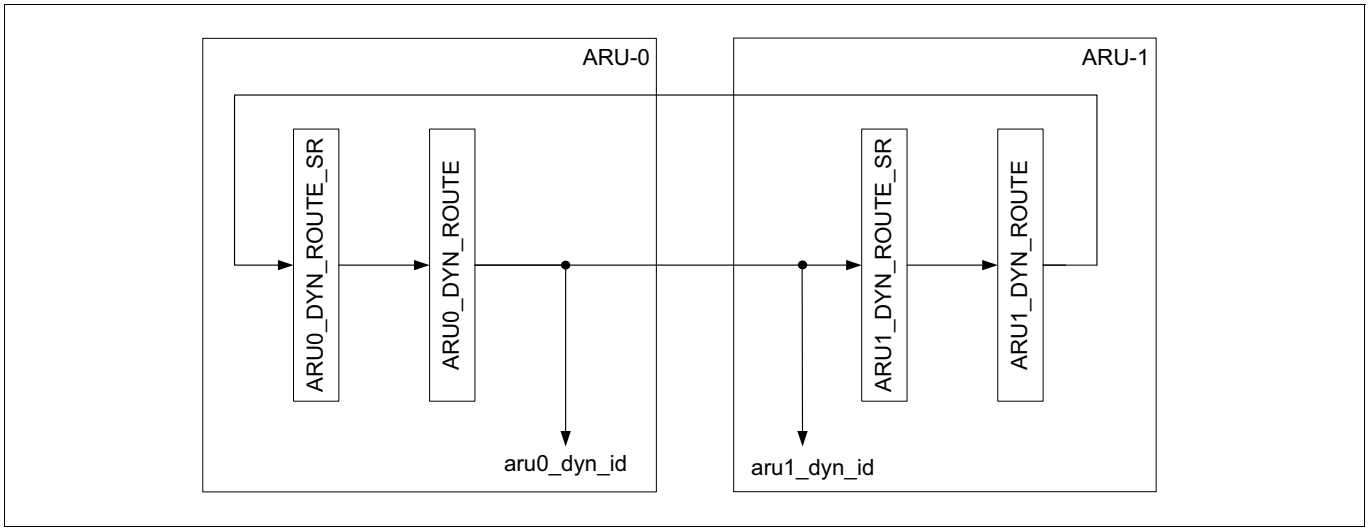


Figure 24 ARU dynamic routing - ring mode

Enabling the dynamic routing ring mode will automatically reset the caddr counter of both ARU-0 and ARU-1. This is necessary to synchronize both ARU's in this mode.

Enabling the dynamic routing ring mode will ignored DYN_ROUTE_SWAP and DYN_UPDATE_EN.

Note: DYN_ARU_UPDATE_EN should be disabled in dynamic routing ring mode.

It is possible to enable the dynamic routing ring mode for both ARU-0 and ARU-1 or only for one of the ARU's by setting the corresponding bit field ARU_0_DYN_EN/ARU_1_DYN_EN of the register **ARU_CTRL.**

Because of the fact that to each ARU port ARU-0 and ARU-1 with the same ARU read ID two different GTM sub-modules are served it may make sense to enable ARU dynamic routing only for one port ARU-0 or ARU-1 if configured to ring-mode. The other port is then served in the default round robin manner.

In dynamic routing ring mode **ARU_[x]_DYN_ROUTE_LOW/_HIGH** and **ARU_[x]_DYN_ROUTE_SR_LOW/_HIGH** are not write-protected. NOTE: Avoid modification of **ARU_[x]_DYN_ROUTE_LOW/_HIGH** and **ARU_[x]_DYN_ROUTE_SR_LOW/_HIGH** in active dynamic routing ring mode.

28.5.4.2 Dynamic routing - ARU controlled

Furthermore it is possible to reload the **ARU_[x]_DYN_ROUTE_SR_LOW/_HIGH** registers by ARU itself.

Therefore the ARU has its own master port which will be served in the normal ARU routing scheme. The ARU read address for this master port has to be configured in the register **ARU_[x]_DYN_RDADDR.**

This feature can be enabled by setting bit DYN_ARU_UPDATE_EN of **ARU_[x]_DYN_CTRL** register.

The following mapping of the ARU word to the **ARU_[x]_DYN_ROUTE_LOW/_HIGH** registers is implemented:

- **ARU_[x]_DYN_ROUTE_SR_LOW(23:0) = aru_data(23:0)**
- **ARU_[x]_DYN_ROUTE_SR_HIGH(28:0) = aru_data(52:24)**

The bit field aru_data(51:48) controls the configuration bits DYN_CLK_WAIT and the bit aru_data(52) controls the configuration bit DYN_UPDATE_EN. Both functions are described in **Section 28.5.4.1**.

In opposite to the dynamic routing scheme controlled from CPU/AEI (only the 6 additional ARU master DYN_REA_ID's are inserted) two additional ID's are served. One is the ARU master ID itself for reloading and the other is the default ID-0. The ID-0 is only added to the inserted routing scheme if bit field **DYN_CLK_WAIT** of **ARU_[x]_DYN_ROUTE_HIGH** is set to zero (only the inserted routing scheme is executed). This ensures that a debug access can take place even if only the inserted routing scheme is executed.

The following dynamic routing scheme is executed for 15 > **DYN_CLK_WAIT** > 0:
… -> ARU-master_ID -> DYN_READ_ID0 ->DYN_READ_ID1 -> DYN_READ_ID2 -> DYN_READ_ID3 -> DYN_READ_ID4 -> DYN_READ_ID5 -> ARU-master_ID -> …

The following dynamic routing scheme is executed for **DYN_CLK_WAIT** = 0:
… -> ARU-master_ID -> DYN_READ_ID0 -> DYN_READ_ID1 -> DYN_READ_ID2 -> DYN_READ_ID3 -> DYN_READ_ID4 -> DYN_READ_ID5 -> default_ID0 -> ARU-master_ID -> …

With the possibility of reloading the dynamic routing scheme over ARU, a FIFO or MCS is able to deliver the dynamic routing scheme data.

28.5.5 ARU Interrupt Signals

Table 13 ARU Interrupt Signals

Signal	Description
ARU_NEW_DATA0_IRQ	Indicates that data is transferred through the ARU using debug channel **ARU_DBG_ACCESS0**.
ARU_NEW_DATA1_IRQ	Indicates that data is transferred through the ARU using debug channel **ARU_DBG_ACCESS1**.
ARU_ACC_ACK_IRQ	ARU access acknowledge IRQ.

28.5.6 ARU Configuration Register Overview

Table 14 ARU Configuration Register Overview

Register name	Description	see Page
ARU_ACCESS	ARU access register	53
ARU_DATA_H	ARU access register upper data word	54
ARU_DATA_L	ARU access register lower data word	55
ARU_DBG_ACCESS0	ARU debug access channel 0	55
ARU_DBG_DATA0_H	ARU debug access 0 transfer register upper data word	57
ARU_DBG_DATA0_L	ARU debug access 0 transfer register lower data word	57
ARU_DBG_ACCESS1	ARU debug access channel 0	58
ARU_DBG_DATA1_H	ARU debug access 1 transfer register upper data word	59
ARU_DBG_DATA1_L	ARU debug access 1 transfer register lower data word	60
ARU_IRQ_NOTIFY	ARU interrupt notification register	60
ARU_IRQ_EN	ARU interrupt enable register	61
ARU_IRQ_FORCINT	ARU force interrupt register	62
ARU_IRQ_MODE	ARU interrupt mode register	63
ARU_CADDR_END	ARU caddr counter end value	63
ARU_CADDR	ARU caddr counter value	64
ARU_CTRL	ARU enable dynamic routing	65
ARU_[z]_DYN_CTRL	ARU z dynamic routing control register	66
ARU_[z]_DYN_RDADDR	ARU z read ID for dynamic routing	66
ARU_[z]_DYN_ROUTE_LOW	ARU z lower bits of DYN_ROUTE register	67
ARU_[z]_DYN_ROUTE_HIGH	ARU z higher bits of DYN_ROUTE register	68
ARU_[z]_DYN_ROUTE_SR_LOW	ARU z shadow register for ARU_[z]_DYN_ROUTE_LOW	68
ARU_[z]_DYN_ROUTE_SR_HIGH	ARU z shadow register for ARU_[z]_DYN_ROUTE_HIGH	69

28.5.7 ARU Configuration Register Description

28.5.7.1 Register ARU_ACCESS

ARU Access Register

Note: The register ARU_ACCESS can be used either for reading or for writing at the same point in time.

ARU_ACCESS
ARU Access Register **(000280$_H$)** **Application Reset Value: 0000 01FE$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		WREQ	RREQ	0			ADDR								
r		rw	rw	r			rw								

Field	Bits	Type	Description
ADDR	8:0	rw	**ARU address** Define the ARU address used for transferring data. For an ARU write request, the preferred address 0x0 have to be used. A write request to the address 0x1FF (always full address) or 0x1FE (always empty address) are ignored and doesn't have any effect. ARU address bits ADDR are only writable if RREQ and WREQ bits are zero.
RREQ	12	rw	**Initiate read request** This bit is cleared automatically after transaction. Moreover, it can be cleared by software to cancel a read request. RREQ bit is only writable if WREQ bit is zero, so to switch from RREQ to WREQ a cancel request has to be performed before. Configuring both RREQ and WREQ bits results in a read request, so RREQ bit will be set if the WREQ bit of the register isn't already set. The ARU read request on address ADDR is served immediately when no other destination has actually a read request when the RREQ bit is set by CPU. In a worst case scenario, the read request is served after one round trip of the ARU, but this is only the case when every destination channel issues a read request at consecutive points in time. 0_B No read request is pending 1_B Set read request to source channel addressed by ADDR

Field	Bits	Type	Description
WREQ	13	rw	**Initiate write request** This bit is cleared automatically after transaction. Moreover, it can be cleared by software to cancel a write request. WREQ bit is only writable if RREQ bit is zero, so to switch from WREQ to RREQ a cancel request has to be performed before. Configuring both RREQ and WREQ bits results in a read request, so WREQ bit will not be set The data is provided at address ADDR. This address has to be programmed as the source address in the destination sub-module channel. In worst case, the data is provided after one full ARU round trip. 0_B No write request is pending 1_B Mark data in registers ARU_DATA_H and ARU_DATA_L as valid
0	11:9, 31:14	r	**Reserved** Read as zero, shall be written as zero.

28.5.7.2 Register ARU_DATA_H

ARU Access Register Upper Data Word

ARU_DATA_H
ARU Access Register Upper Data Word **(000284_H)** **Application Reset Value: 0000 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0			**DATA**												
r			rw												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA															
rw															

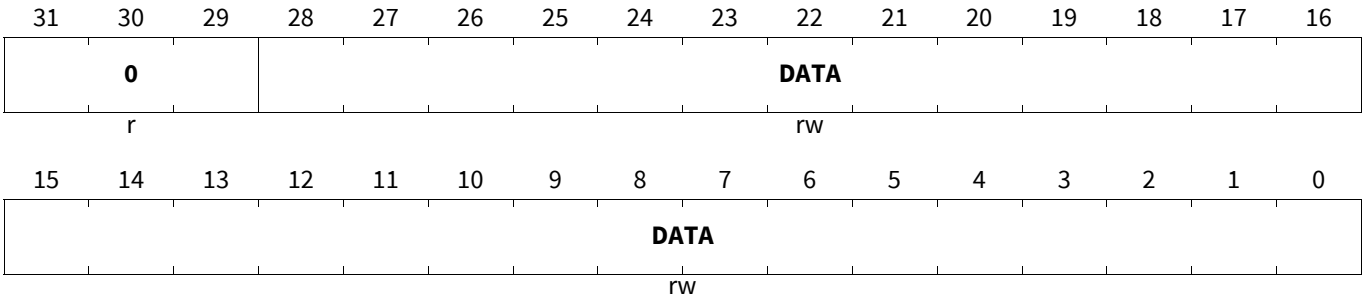
Field	Bits	Type	Description
DATA	28:0	rw	**Upper ARU data word** Transfer upper ARU data word addressed by ADDR. The data bits 24 to 52 of an ARU word are mapped to the data bits 0 to 28 of this register.
0	31:29	r	**Reserved** Read as zero, shall be written as zero.

28.5.7.3 Register ARU_DATA_L

ARU Access Register Lower Data Word

ARU_DATA_L

ARU Access Register Lower Data Word **(000288$_H$)** **Application Reset Value: 0000 0000$_H$**



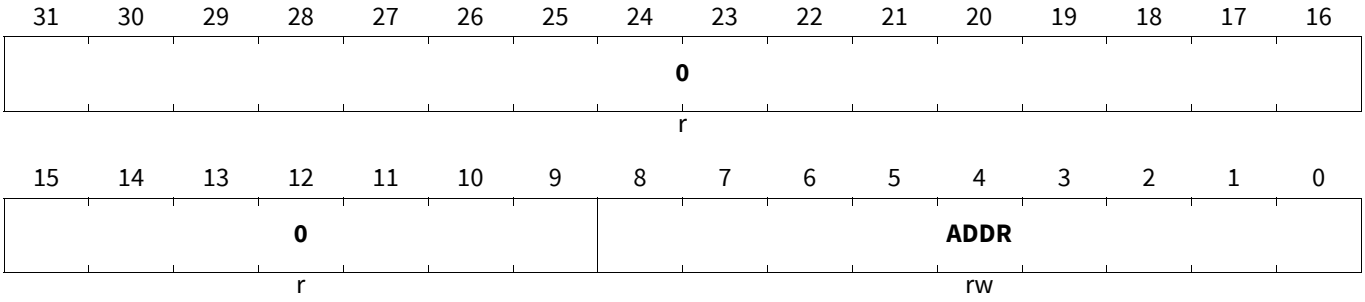
Field	Bits	Type	Description
DATA	28:0	rw	**Lower ARU data word** Transfer lower ARU data word addressed by ADDR. The data bits 0 to 23 of an ARU word are mapped to the data bits 0 to 23 of this register and the data bits 48 to 52 of an ARU word are mapped to the data bits 24 to 28 of this register when data is read by the CPU. For writing data into the ARU by the CPU the bits 24 to 28 are **not** transferred to bit 48 to 52 of the ARU word. Only bits 0 to 23 are written to bits 0 to 23 of the ARU word.
0	31:29	r	**Reserved** Read as zero, shall be written as zero.

28.5.7.4 Register ARU_DBG_ACCESS0

ARU Debug Access Channel 0

ARU_DBG_ACCESS0

ARU Debug Access Channel 0 **(00028C$_H$)** **Application Reset Value: 0000 01FE$_H$**



Field	Bits	Type	Description
ADDR	8:0	rw	**ARU debugging address** Define address of ARU debugging channel 0.

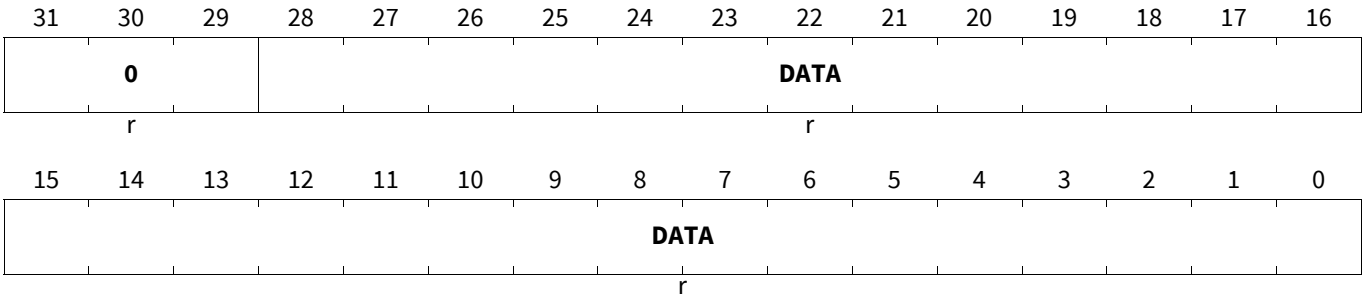
Field	Bits	Type	Description
0	31:9	r	**Reserved** Read as zero, shall be written as zero.

28.5.7.5 Register ARU_DBG_DATA0_H

ARU Debug Access 0 Transfer Register Upper Data Word

ARU_DBG_DATA0_H

ARU Debug Access 0 Transfer Register Upper Data Word(000290$_H$) **Application Reset Value: 0000 0000$_H$**



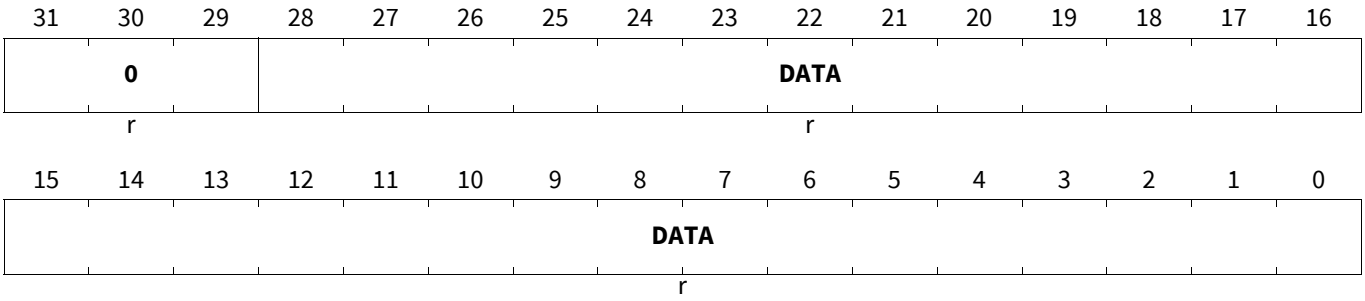
Field	Bits	Type	Description
DATA	28:0	r	**Upper debug data word** Transfer upper ARU data word addressed by register **DBG_ACCESS0**. The data bits 24 to 52 of an ARU word are mapped to the data bits 0 to 28 of this register The interrupt *ARU_NEW_DATA0_IRQ* is raised if a new data word is available.
0	31:29	r	**Reserved** Read as zero, shall be written as zero.

28.5.7.6 Register ARU_DBG_DATA0_L

ARU Debug Access 0 Transfer Register Lower Data Word

ARU_DBG_DATA0_L

ARU Debug Access 0 Transfer Register Lower Data Word(000294$_H$) **Application Reset Value: 0000 0000$_H$**



Field	Bits	Type	Description
DATA	28:0	r	**Lower debug data word** Transfer lower ARU data word addressed by register **DBG_ACCESS0**. The data bits 0 to 23 of an ARU word are mapped to the data bits 0 to 23 of this register and the data bits 48 to 52 of an ARU word is mapped to the data bits 24 to 28 of this register. The interrupt *ARU_NEW_DATA0_IRQ* is raised if a new data word is available.
0	31:29	r	**Reserved** Read as zero, shall be written as zero

28.5.7.7 Register ARU_DBG_ACCESS1

ARU Debug Access Channel 1

ARU_DBG_ACCESS1
ARU Debug Access Channel 1 **(000298_H)** **Application Reset Value: 0000 01FE$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0							**ADDR**								
r							rw								

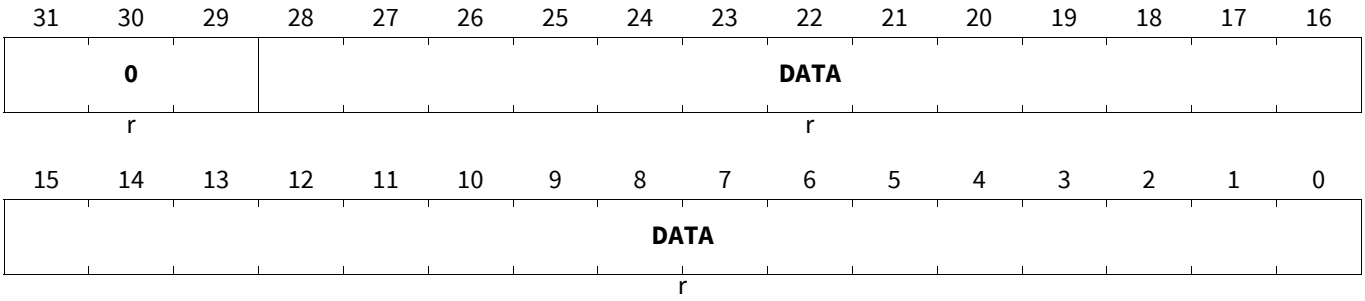
Field	Bits	Type	Description
ADDR	8:0	rw	**ARU debugging address** Define address of ARU debugging channel 1.
0	31:9	r	**Reserved** Read as zero, shall be written as zero

28.5.7.8 Register ARU_DBG_DATA1_H

ARU Debug Access 1 Transfer Register Upper Data Word

ARU_DBG_DATA1_H

ARU Debug Access 1 Transfer Register Upper Data Word(00029C$_H$) **Application Reset Value: 0000 0000$_H$**



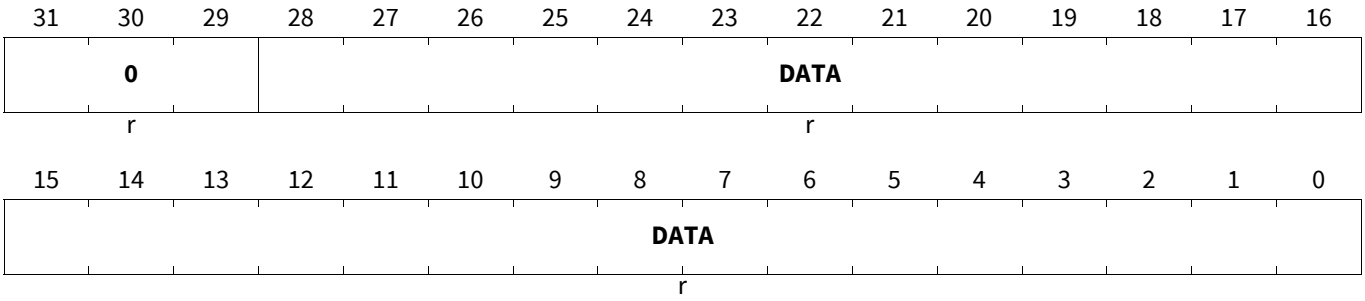
Field	Bits	Type	Description
DATA	28:0	r	**Upper debug data word** Transfer upper ARU data word addressed by register **DBG_ACCESS1**. The data bits 24 to 52 of an ARU word are mapped to the data bits 0 to 28 of this register. The interrupt *ARU_NEW_DATA1_IRQ* is raised if a new data word is available.
0	31:29	r	**Reserved** Read as zero, shall be written as zero.

28.5.7.9 Register ARU_DBG_DATA1_L

ARU Debug Access 1 Transfer Register Lower Data Word

ARU_DBG_DATA1_L

ARU Debug Access 1 Transfer Register Lower Data Word(0002A0$_H$) **Application Reset Value: 0000 0000$_H$**



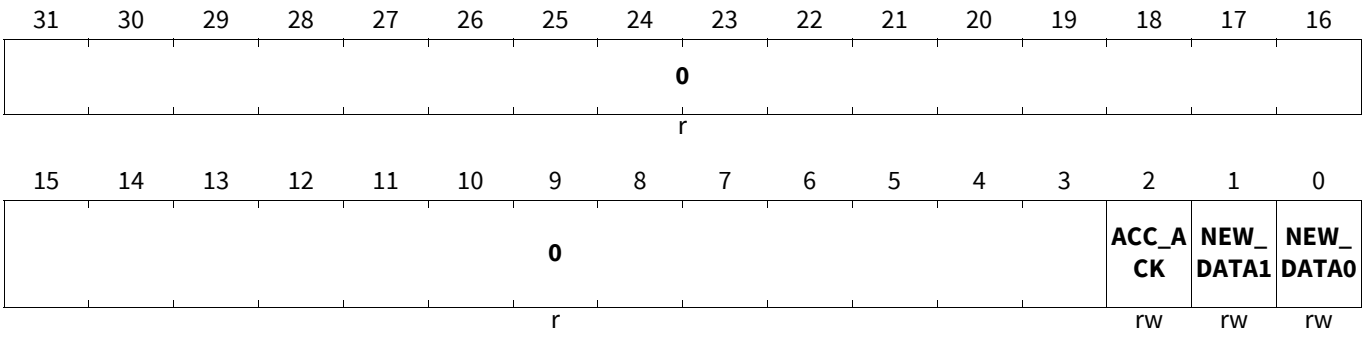
Field	Bits	Type	Description
DATA	28:0	r	**Lower debug data word** Transfer lower ARU data word addressed by register **DBG_ACCESS1**. The data bits 0 to 23 of an ARU word are mapped to the data bits 0 to 23 of this register and the data bits 48 to 52 of an ARU word is mapped to the data bits 24 to 28 of this register. The interrupt *ARU_NEW_DATA1_IRQ* is raised if a new data word is available.
0	31:29	r	**Reserved** Read as zero, shall be written as zero.

28.5.7.10 Register ARU_IRQ_NOTIFY

ARU Interrupt Notification Register

ARU_IRQ_NOTIFY

ARU Interrupt Notification Register **(0002A4$_H$)** **Application Reset Value: 0000 0000$_H$**

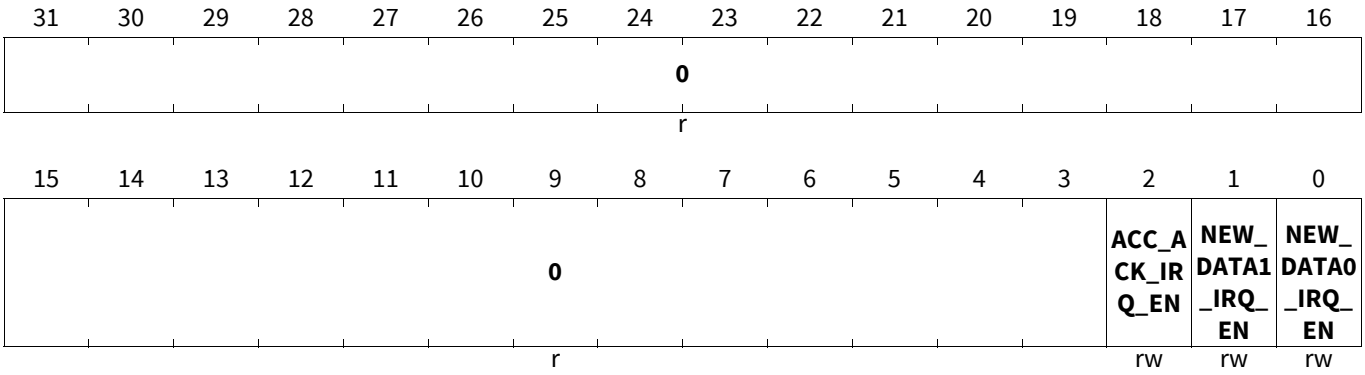


Field	Bits	Type	Description
NEW_DATA0	0	rw	**Data was transferred for addr ARU_DBG_ACCESS0** This bit will be cleared on a CPU write access of value '1'. (As the bit is rw, otherwise no clear.) A read access leaves the bit unchanged. 0_B No interrupt occurred 1_B ARU_NEW_DATA0_IRQ interrupt was raised by the ARU
NEW_DATA1	1	rw	**Data was transferred for addr ARU_DBG_ACCESS1** This bit will be cleared on a CPU write access of value '1'. (As the bit is rw, otherwise no clear.) A read access leaves the bit unchanged. 0_B No interrupt occurred 1_B ARU_NEW_DATA1_IRQ interrupt was raised by the ARU
ACC_ACK	2	rw	**AEI to ARU access finished, on read access data are valid** This bit will be cleared on a CPU write access of value '1'. (As the bit is rw, otherwise no clear.) A read access leaves the bit unchanged.
0	31:3	r	**Reserved** Read as zero, shall be written as zero.

28.5.7.11 Register ARU_IRQ_EN

ARU Interrupt Enable Register

ARU_IRQ_EN
ARU Interrupt Enable Register **($0002A8_H$)** **Application Reset Value: $0000\ 0000_H$**



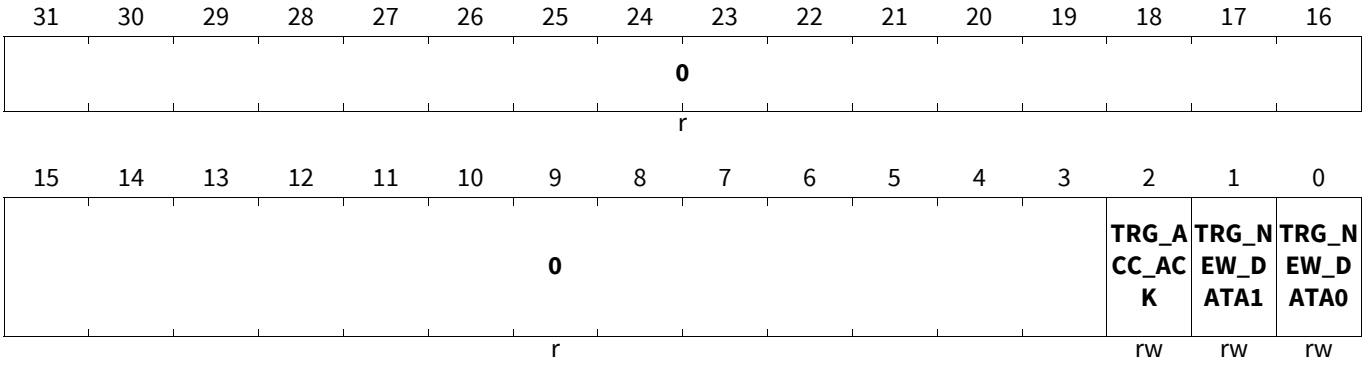
Field	Bits	Type	Description
NEW_DATA0_IRQ_EN	0	rw	**ARU_NEW_DATA0_IRQ interrupt enable** 0_B Disable interrupt, interrupt is not visible outside GTM 1_B Enable interrupt, interrupt is visible outside GTM
NEW_DATA1_IRQ_EN	1	rw	**ARU_NEW_DATA1_IRQ interrupt enable** 0_B Disable interrupt, interrupt is not visible outside GTM 1_B Enable interrupt, interrupt is visible outside GTM
ACC_ACK_IRQ_EN	2	rw	**ACC_ACK_IRQ interrupt enable** 0_B Disable interrupt, interrupt is not visible outside GTM 1_B Enable interrupt, interrupt is visible outside GTM

Field	Bits	Type	Description
0	31:3	r	**Reserved** Read as zero, shall be written as zero.

28.5.7.12 Register ARU_IRQ_FORCINT

ARU Force Interrupt Register

ARU_IRQ_FORCINT
ARU Force Interrupt Register **(0002AC$_H$)** **Application Reset Value: 0000 0000$_H$**

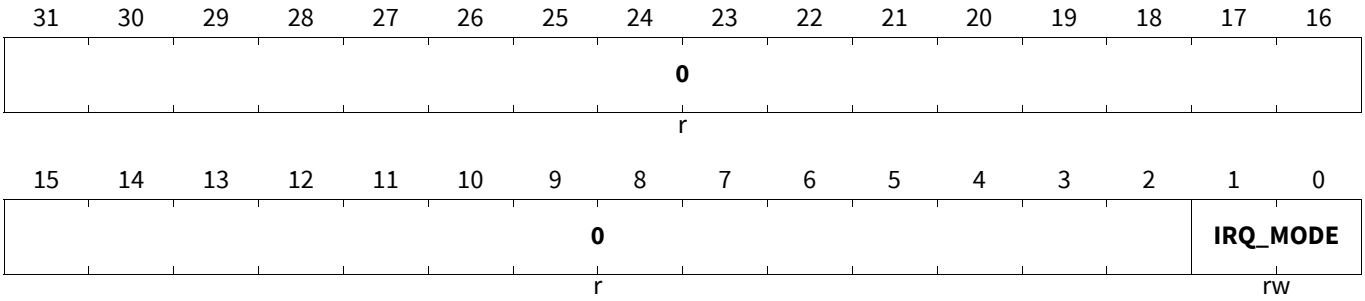


Field	Bits	Type	Description
TRG_NEW_DATA0	0	rw	**Trigger new data 0 interrupt** Note: This bit is cleared automatically after write. This bit is write protected by bit RF_PROT of register GTM_CTRL. 0_B Corresponding bit in status register will not be forced 1_B Assert corresponding field in ARU_IRQ_NOTIFY register
TRG_NEW_DATA1	1	rw	**Trigger new data 1 interrupt** Note: This bit is cleared automatically after write. This bit is write protected by bit RF_PROT of register GTM_CTRL. 0_B Corresponding bit in status register will not be forced 1_B Assert corresponding field in ARU_IRQ_NOTIFY register
TRG_ACC_ACK	2	rw	**Trigger ACC_ACK interrupt** Note: This bit is cleared automatically after write. This bit is write protected by bit RF_PROT of register GTM_CTRL. 0_B Corresponding bit in status register will not be forced 1_B Assert corresponding field in ARU_IRQ_NOTIFY register
0	31:3	r	**Reserved** Read as zero, shall be written as zero.

28.5.7.13 Register ARU_IRQ_MODE

ARU Interrupt Mode Register

ARU_IRQ_MODE
ARU Interrupt Mode Register **(0002B0$_H$)** **Application Reset Value: 0000 0000$_H$**

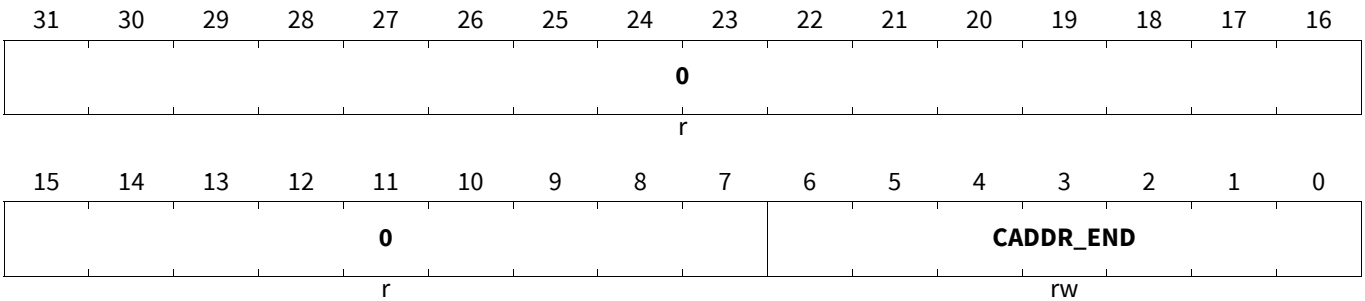


Field	Bits	Type	Description
IRQ_MODE	1:0	rw	**IRQ mode selection** The interrupt modes are described in **Section 28.4.5**. 00_B Level mode 01_B Pulse mode 10_B Pulse-Notify mode 11_B Single-Pulse mode
0	31:2	r	**Reserved** Read as zero, shall be written as zero.

28.5.7.14 Register ARU_CADDR_END

ARU caddr Counter End Value Register

ARU_CADDR_END
ARU caddr Counter End Value Register **(0002B4$_H$)** **Application Reset Value: 0000 007F$_H$**



Field	Bits	Type	Description
CADDR_END	6:0	rw	**Set end value of ARU caddr counter** The ARU roundtrip counter aru_caddr runs from zero to caddr_end value. Shorten the ARU roundtrip cycle by setting a smaller number than the defined reset value will cause that not all ARU-connected modules will be served. Making the roundtrip cycle longer than the reset value would cause longer ARU roundtrip time and as a result some ARU-connected modules will not be served as fast as possible for this device. This bit is write protected by bit RF_PROT of register GTM_CTRL
0	31:7	r	**Reserved** Read as zero, shall be written as zero.

28.5.7.15 Register ARU_CADDR

ARU caddr Counter Value

Note: The registers CADDR_0 and CADDR_1 start incrementing with each clock cycle just after reset. Due to this the initial reset value cannot be read back.

ARU_CADDR

ARU caddr Counter Value **(0002FC$_H$)** **Application Reset Value: 0000 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0									**CADDR_1**						
r									r						

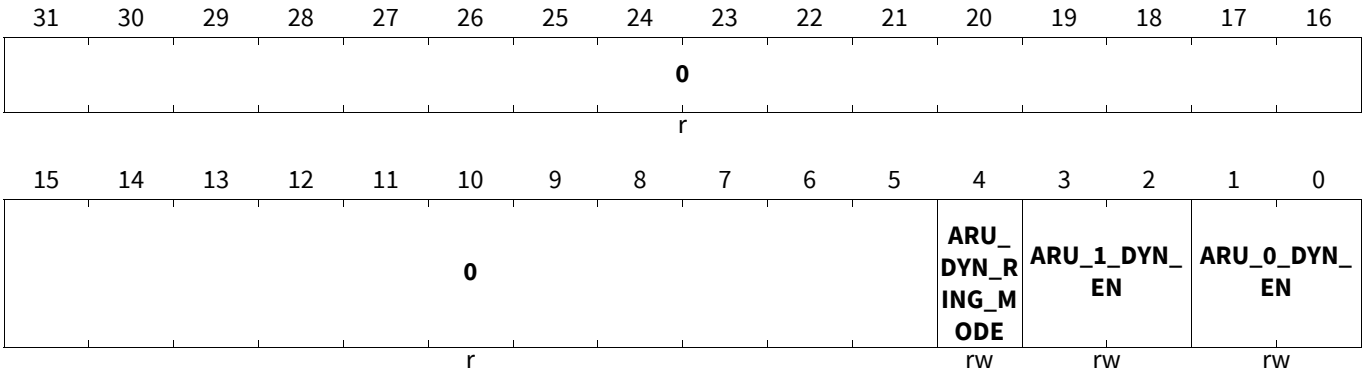
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0									**CADDR_0**						
r									r						

Field	Bits	Type	Description
CADDR_0	6:0	r	**Value of ARU-0 caddr counter**
CADDR_1	22:16	r	**Value of ARU-1 caddr counter**
0	15:7, 31:23	r	**Reserved** Read as zero, shall be written as zero.

28.5.7.16 Register ARU_CTRL

ARU Enable Dynamic Routing Register

ARU_CTRL
ARU Enable Dynamic Routing Register (0002BC$_H$) Application Reset Value: 0000 0000$_H$

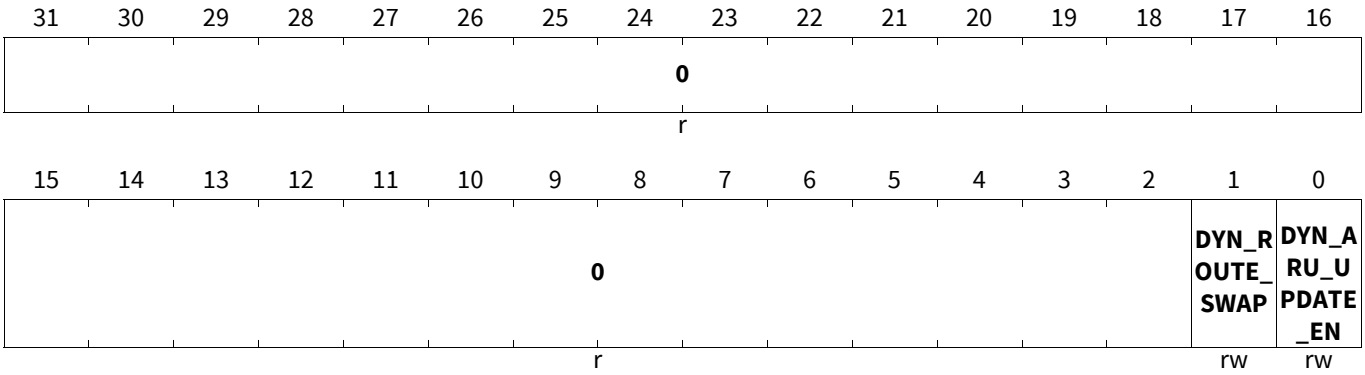


Field	Bits	Type	Description
ARU_0_DYN_EN	1:0	rw	**Enable dynamic routing for ARU-0** Dynamic routing enable of ARU-0. Write of following double bit values is possible: If dynamic routing is disabled, the normal ARU routing scheme for ARU-0 is executed. 00_B no change 01_B Disable dynamic routing 10_B Enable dynamic routing 11_B no change
ARU_1_DYN_EN	3:2	rw	**Enable dynamic routing for ARU-1** Dynamic routing enable of ARU-1. Write of following double bit values is possible: If dynamic routing is disabled, the normal ARU routing scheme for ARU-1 is executed. 00_B no change 01_B Disable dynamic routing 10_B Enable dynamic routing 11_B no change
ARU_DYN_RING_MODE	4	rw	**Enable dynamic routing ring mode** Dynamic routing ring mode for both ARU-0 and ARU-1. 0_B Different dynamic routing scheme for ARU-0 and ARU-11 1_B Same dynamic routing scheme for ARU-0 and ARU-1 with 24 possible read-ID's (dynamic routing ring mode)
0	31:5	r	**Reserved** Read as zero, shall be written as zero.

28.5.7.17 Register ARU_[z]_DYN_CTRL

ARU z Dynamic Routing Control Register

ARU_z_DYN_CTRL (z=0-1)
ARU z Dynamic Routing Control Register (0002C0$_H$+z*4) Application Reset Value: 0000 0000$_H$

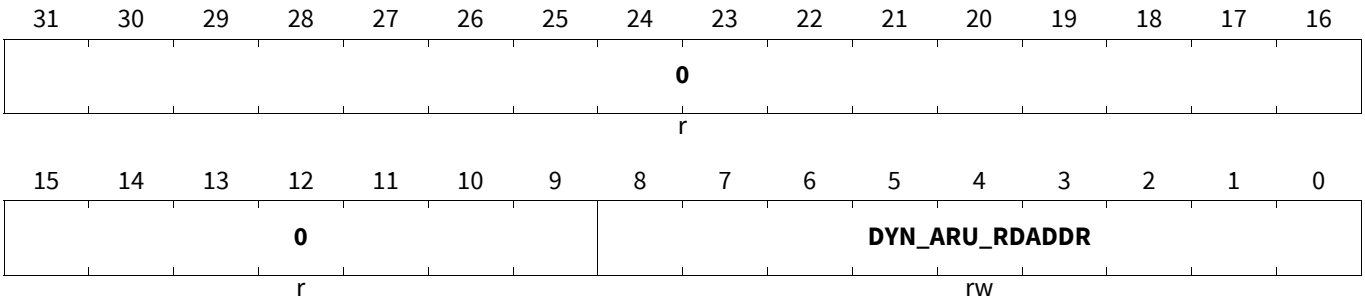


Field	Bits	Type	Description
DYN_ARU_UPDATE_EN	0	rw	**Enable reload of DYN_ROUTE register from ARU itself** Enable reload of DYN_ROUTE register from ARU itself.
DYN_ROUTE_SWAP	1	rw	**Enable swapping DYN_ROUTE_SR with DYN_ROUTE register** Enable swapping **DYN_ROUTE_SR** with **DYN_ROUTE** register.
0	31:2	r	**Reserved** Read as zero, shall be written as zero.

28.5.7.18 Register ARU_[z]_DYN_RDADDR

ARU z Read ID for Dynamic Routing

ARU_z_DYN_RDADDR (z=0-1)
ARU z Read ID for Dynamic Routing (0002E8$_H$+z*4) Application Reset Value: 0000 0000$_H$



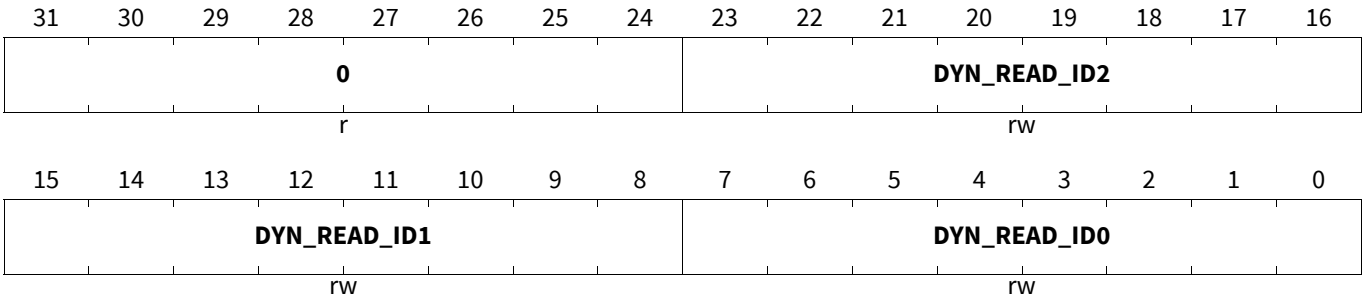
Field	Bits	Type	Description
DYN_ARU_RDADDR	8:0	rw	**ARU read address ID to reload the DYN_ROUTE register** ARU read address ID to reload the DYN_ROUTE register from ARU itself.
0	31:9	r	**Reserved** Read as zero, shall be written as zero.

28.5.7.19 Register ARU_[z]_DYN_ROUTE_LOW

ARU z Lower Bits of DYN_ROUTE Register

ARU_z_DYN_ROUTE_LOW (z=0-1)

ARU z Lower Bits of DYN_ROUTE Register (0002C8$_H$+z*4) Application Reset Value: 0000 0000$_H$



Field	Bits	Type	Description
DYN_READ_ID0	7:0	rw	**ARU read ID 0** ARU read ID 0 for dynamic routing.
DYN_READ_ID1	15:8	rw	**ARU read ID 2** ARU read ID 1 for dynamic routing.
DYN_READ_ID2	23:16	rw	**ARU read ID 2** ARU read ID 2 for dynamic routing.
0	31:24	r	**Reserved** Read as zero, shall be written as zero

28.5.7.20 Register ARU_[z]_DYN_ROUTE_HIGH

ARU z Higher Bits of DYN_ROUTE Register

ARU_z_DYN_ROUTE_HIGH (z=0-1)

ARU z Higher Bits of DYN_ROUTE Register (0002D0$_H$+z*4) Application Reset Value: 0000 0000$_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0				DYN_CLK_WAIT				DYN_READ_ID5							
r				rw				rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DYN_READ_ID4								DYN_READ_ID3							
rw								rw							

Field	Bits	Type	Description
DYN_READ_ID3	7:0	rw	**ARU read ID 3** ARU read ID 3 for dynamic routing.
DYN_READ_ID4	15:8	rw	**ARU read ID 4** ARU read ID 4 for dynamic routing.
DYN_READ_ID5	23:16	rw	**ARU read ID 5** ARU read ID 5 for dynamic routing.
DYN_CLK_WAIT	27:24	rw	**Number of clk cycles for dynamic routing** Defines the number of clk cycles between each dynamic routing ID.
0	31:28	r	**Reserved** Read as zero, shall be written as zero.

28.5.7.21 Register ARU_[z]_DYN_ROUTE_SR_LOW

ARU z Shadow Register for ARU_z_DYN_ROUTE_LOW

NOTE : This is the shadow register for register **ARU_[z]_DYN_ROUTE_LOW**

ARU_z_DYN_ROUTE_SR_LOW (z=0-1)

ARU z Shadow Register for ARU_z_DYN_ROUTE_LOW(0002D8$_H$+z*4) Application Reset Value: 0000 0000$_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								DYN_READ_ID8							
r								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DYN_READ_ID7								DYN_READ_ID6							
rw								rw							

Field	Bits	Type	Description
DYN_READ_ID6	7:0	rw	**ARU read ID 6** ARU read ID 6 for dynamic routing. These bits are mapped to ARU data bits aru_data(7:0).
DYN_READ_ID7	15:8	rw	**ARU read ID 7** ARU read ID 7 for dynamic routing. These bits are mapped to ARU data bits aru_data(15:8).
DYN_READ_ID8	23:16	rw	**ARU read ID 8** ARU read ID 8 for dynamic routing. These bits are mapped to ARU data bits aru_data(23:16).
0	31:24	r	**Reserved** Read as zero, shall be written as zero.

28.5.7.22 Register ARU_[z]_DYN_ROUTE_SR_HIGH

ARU z Shadow Register for ARU_z_DYN_ROUTE_HIGH

NOTE : This is the shadow register for register **ARU_[z]_DYN_ROUTE_HIGH**

ARU_z_DYN_ROUTE_SR_HIGH (z=0-1)

ARU z Shadow Register for ARU_z_DYN_ROUTE_HIGH(0002E0$_H$+z*4) Application Reset Value: 0000 0000$_H$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0			DYN_UPDATE_EN	DYN_CLK_WAIT				DYN_READ_ID11							
r			rw	rw				rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DYN_READ_ID10								DYN_READ_ID9							
rw								rw							

Field	Bits	Type	Description
DYN_READ_ID9	7:0	rw	**ARU read ID 9** ARU read ID 9 for dynamic routing. These bits are mapped to ARU data bits aru_data(31:24).
DYN_READ_ID10	15:8	rw	**ARU read ID 10** ARU read ID 10 for dynamic routing. These bits are mapped to ARU data bits aru_data(39:32).
DYN_READ_ID11	23:16	rw	**ARU read ID 11** ARU read ID 11 for dynamic routing. These bits are mapped to ARU data bits aru_data(47:40).
DYN_CLK_WAIT	27:24	rw	**Number of clk cycles for dynamic routing** Defines the number of clk cycles between each dynamic routing ID. These bits are mapped to ARU data bits aru_data(51:48).

Field	Bits	Type	Description
DYN_UPDATE_EN	28	rw	**Update enable from shadow register** Enable update **ARU_[z]_DYN_ROUTE_LOW/_HIGH** registers from shadow registers **ARU_[z]_DYN_ROUTE_SR_LOW/_HIGH**. This bit is mapped to ARU data bit aru_data(52).
0	31:29	r	**Reserved** Read as zero, shall be written as zero.