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.