DMA
Direct Memory Access

AURIX™ TC2xx Microcontroller Training
V1.0 2019-03

DMA
Direct Memory Access
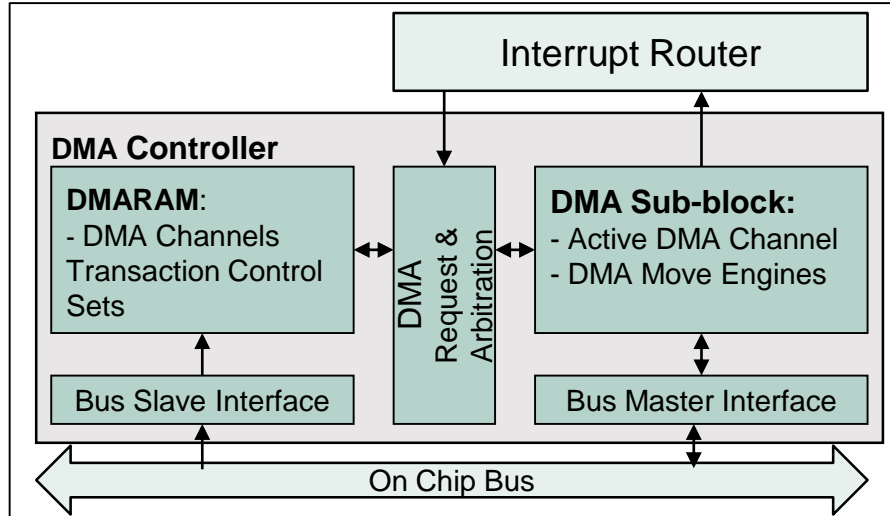
Highlights

› The DMA moves data from source locations to destination locations without the intervention of the CPU or other on chip devices.

› Up to 128 individually programmable DMA channels

Key Features

Flexible DMA channels requests

DMA double buffering

DMA linked list

Customer Benefits

› Possible configuration of the request type (SW, HW, Auto, ..) per DMA channel

› Transfer continuous data stream to two destination buffers

› Perform multi DMA transactions from non contiguous Memory regions

DMA
Flexible DMA channels requests

› The DMA Channel supports the following types of requests:

  – **DMA Software Request**: initiated by CPU

  – **DMA Hardware Request**: Any peripheral that can trigger an interrupt can initiate a DMA transaction through the Interrupt Router

  – **DMA Daisy Chain Request**: DMA transaction initiated by the next higher priority DMA channel

  – **DMA Auto Start Request:** initiated by the loading of the next Transaction Control Set (TCS) during a DMA Linked List operation
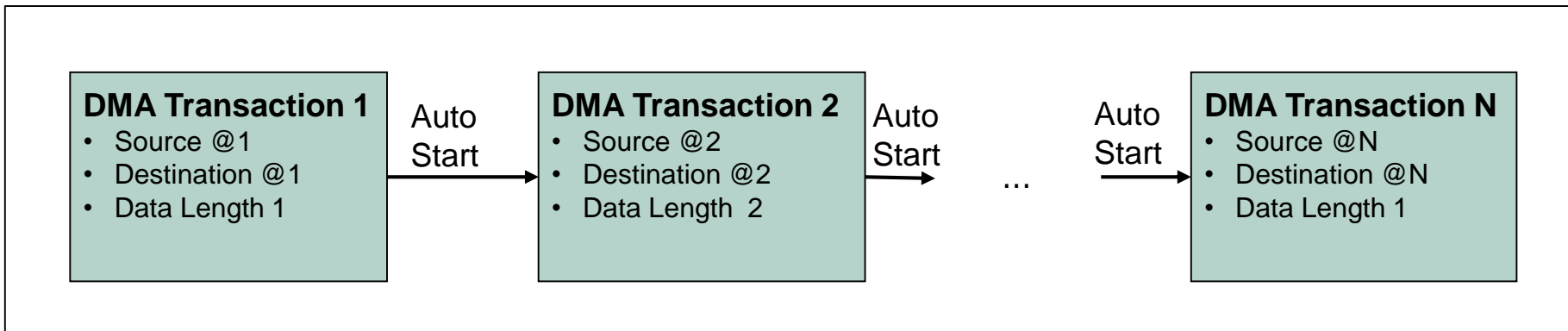
DMA
DMA double buffering

› Double buffer could be selected for source or destination buffering

› The application is able to freeze one of the destination buffers for cyclic software tasks while the other buffer continues to be filled

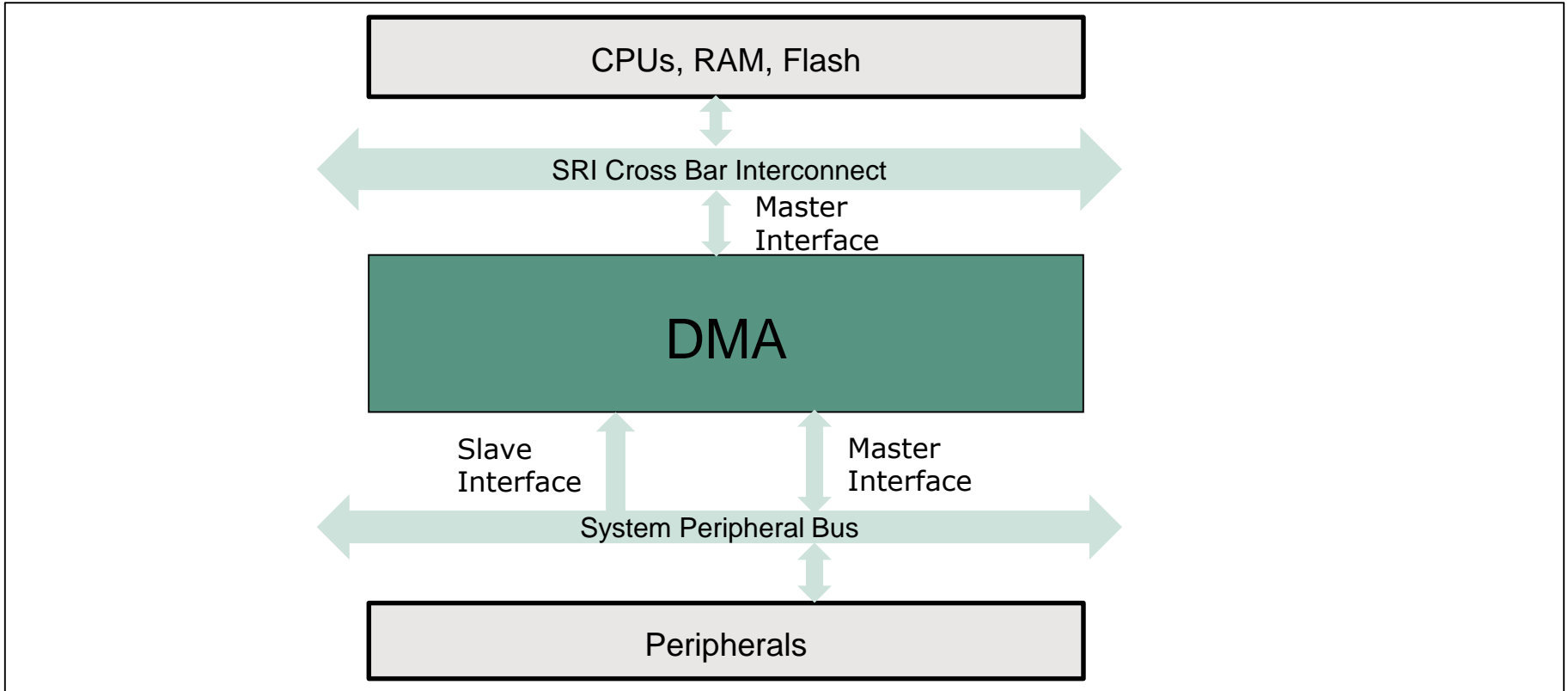

Double Destination Buffering
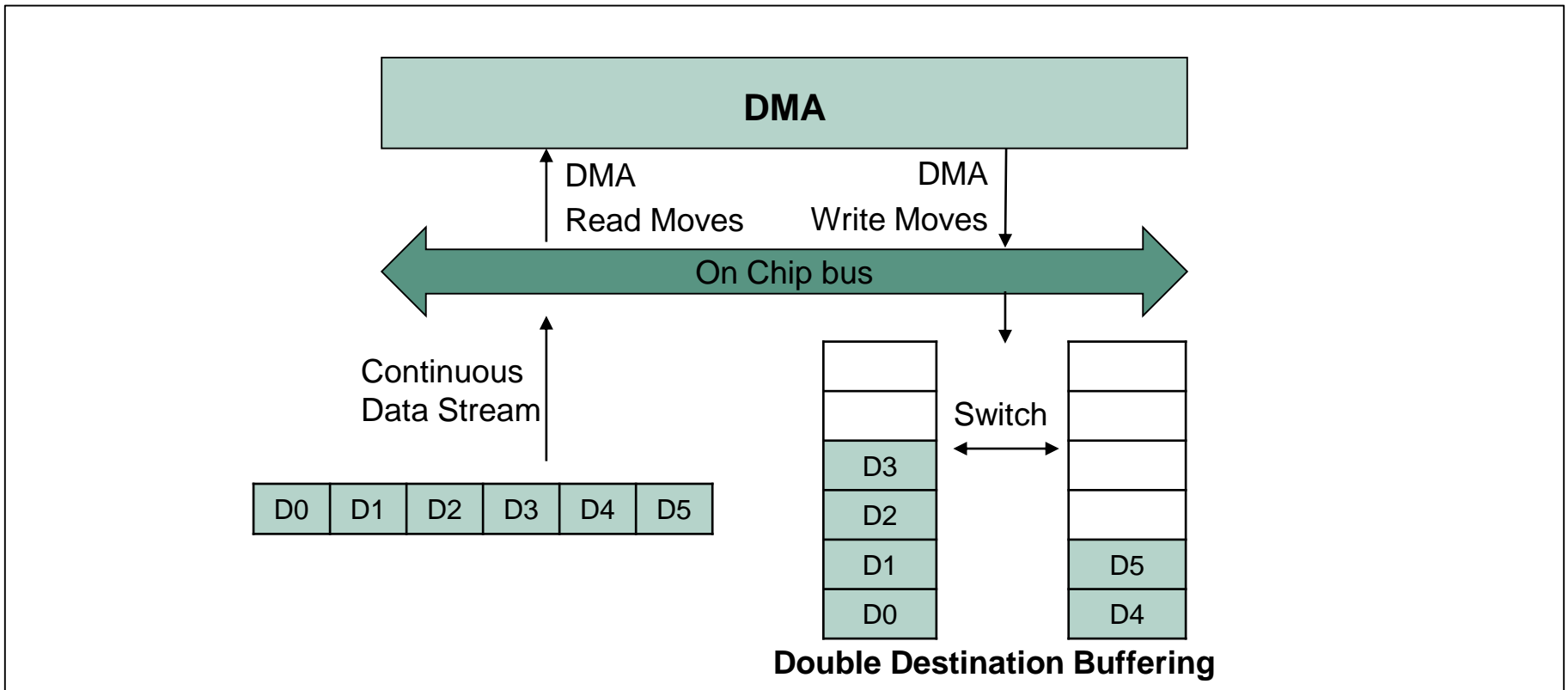
DMA
DMA linked list

› A linked list operation consists of a series of DMA transactions executed by **the same DMA channel**. Each DMA transaction has an unique configuration Set

› If the Auto start request is selected, a DMA transaction will be triggered after the end of the previous one in the list ➔ no HW or SW trigger is needed

DMA Transaction 1	Auto Start	DMA Transaction 2	Auto Start	...	Auto Start	DMA Transaction N
• Source @1 • Destination @1 • Data Length 1		• Source @2 • Destination @2 • Data Length 2				• Source @N • Destination @N • Data Length 1

› DMA linked are useful when user wants to transfer data from/to non contiguous memory locations using **one** DMA channel and/or **one** service request
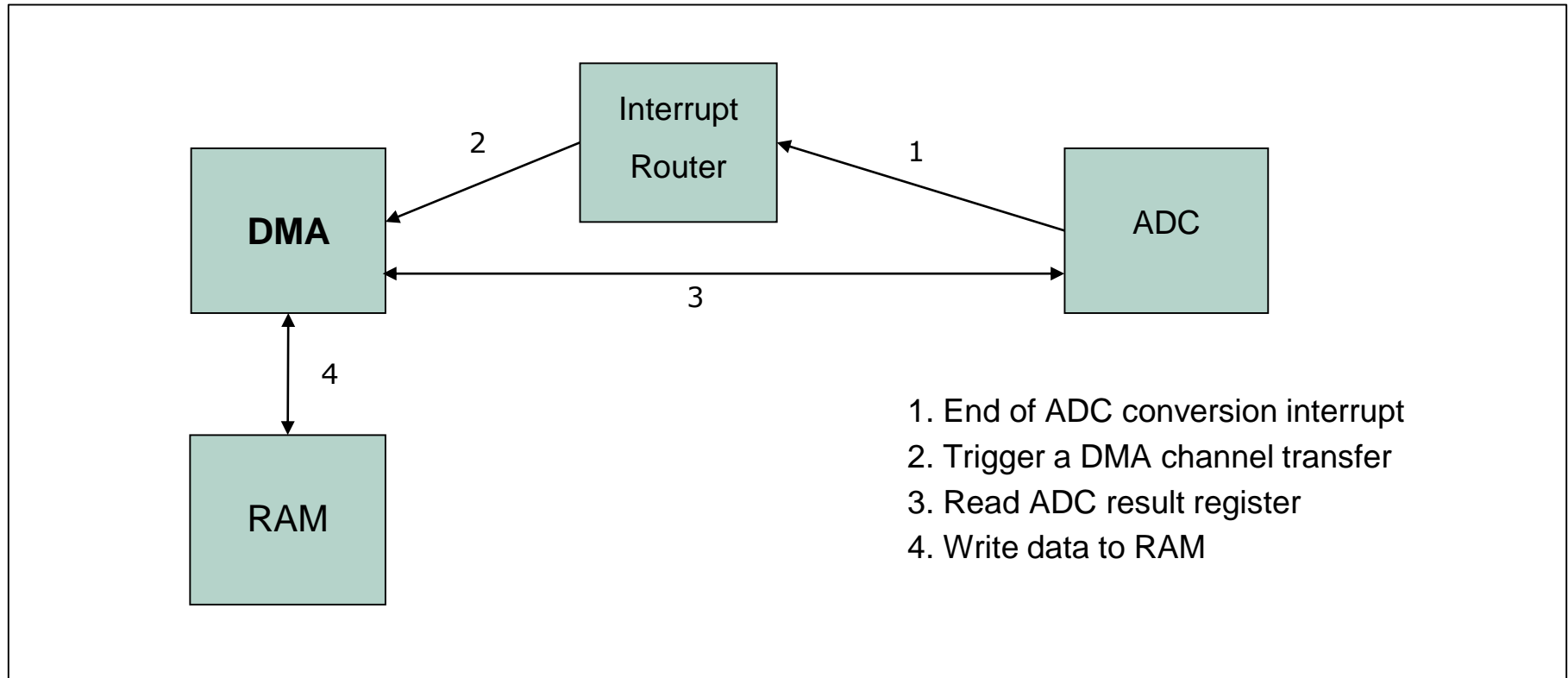
DMA
System integration


```
┌─────────────────────────────────────────────────────────┐
│                   CPUs, RAM, Flash                      │
└─────────────────────────────────────────────────────────┘
            ↕
◄──────────── SRI Cross Bar Interconnect ────────────►
                        ↕  Master
                           Interface
         ┌───────────────────────────────────┐
         │               DMA                 │
         └───────────────────────────────────┘
            ↑ Slave          ↕ Master
              Interface         Interface
◄──────────── System Peripheral Bus ────────────►
                        ↕
         ┌───────────────────────────────────┐
         │            Peripherals            │
         └───────────────────────────────────┘
```

› The DMA is connected to both SRI and SPB with master interfaces

› This enables the DMA to read and write data from/to any module

› In this example, data is transferred from the ADC output registers to internal memory without any CPU intervention

1. End of ADC conversion interrupt
2. Trigger a DMA channel transfer
3. Read ADC result register
4. Write data to RAM

Edition 2019-03
Published by
Infineon Technologies AG
81726 Munich, Germany

© 2019 Infineon Technologies AG.
All Rights Reserved.

Do you have a question about this document?
Email: erratum@infineon.com

Document reference
AURIX_Training_1_
Direct_Memory_Access