



SAM4S Peripheral DMA Controller (PDC)

Presentation Outline

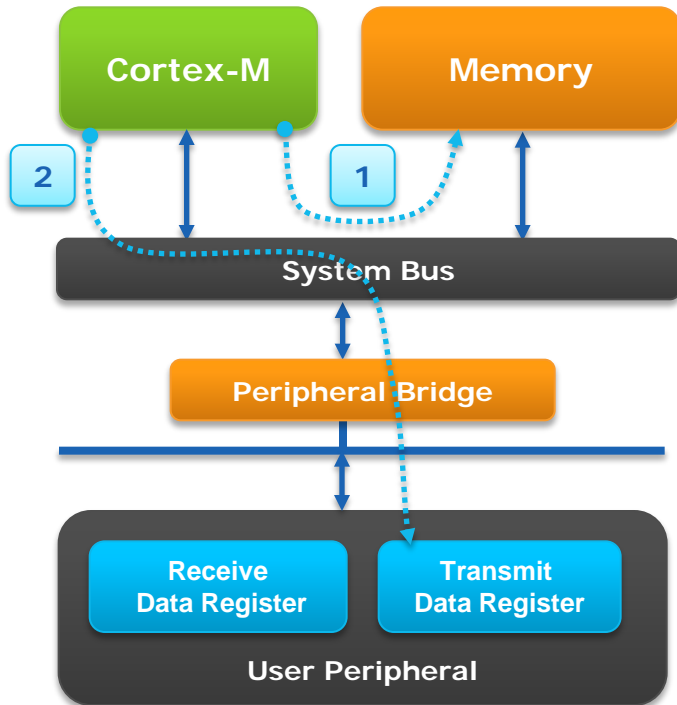
- Introduction
- Functional Description
- Application Examples

Introduction

Introduction

Data Transfer without DMA

- Transmit Transfer Example:

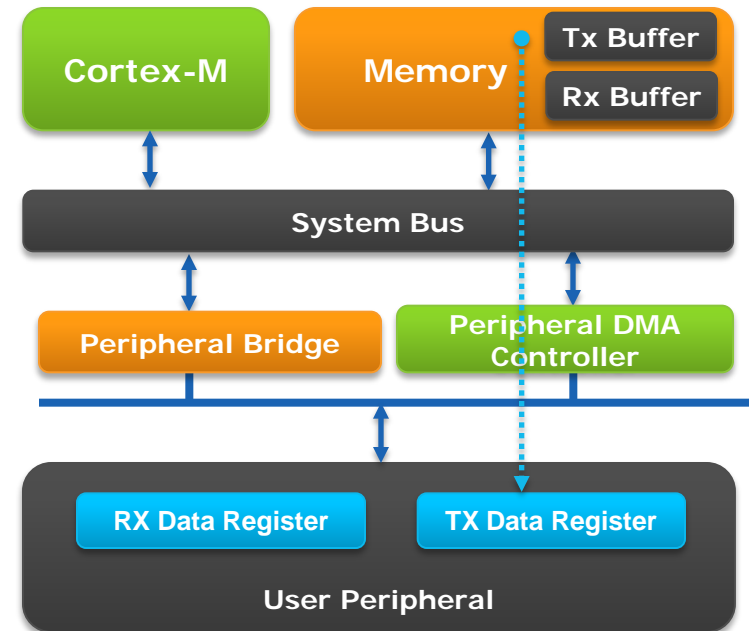


- Each Read or Write accesses to Receive or Transmit Registers must be handled by the core
 1. Data read from Memory
 2. Data write to Transmit Data Register
- CPU spends most of its time transferring large amount of data

Introduction

Data Transfer using Peripheral DMA Controller (PDC)

- Each Read or Write accesses to Receive or Transmit Registers are handled by the PDC
 - 1 clock cycle for a transfer from memory to peripheral
 - 2 clock cycles for a transfer from peripheral to memory
- No need for the CPU to handle data transfer
 - Free for other tasks
 - Can be disabled to reduce power consumption

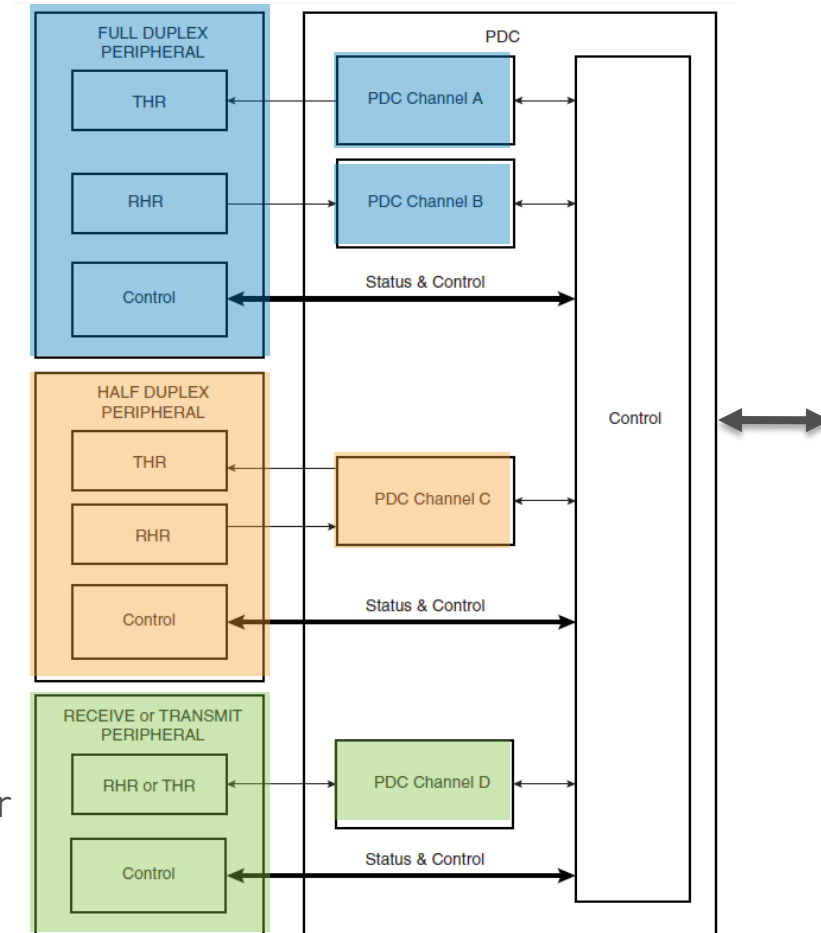


Functional Description

Functional Description

Overview

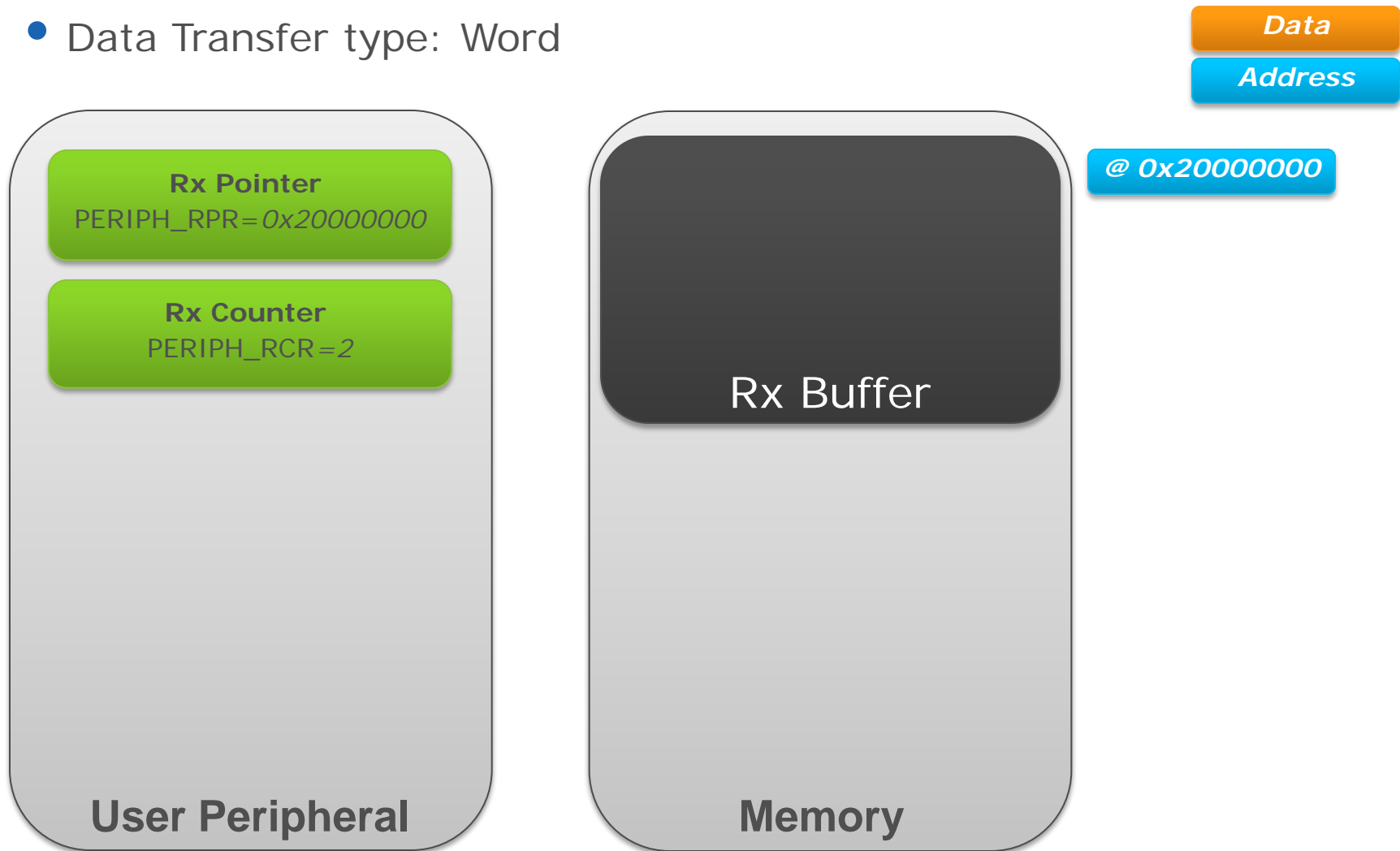
- The PDC transfers data between:
 - the on-chip peripherals
 - and the on-/off-chip memories
- Features
 - One AHB Master
 - Transfers without CPU intervention
 - Periph to Mem & Mem to Periph transfers
 - Simplex, Half and Full Duplex
 - 8-bit, 16-bit and 32-bit data transfers
 - Buffer chaining for continuous Data transfer
 - Transfers without CPU intervention



Functional Description

Simple PDC Data Transfer

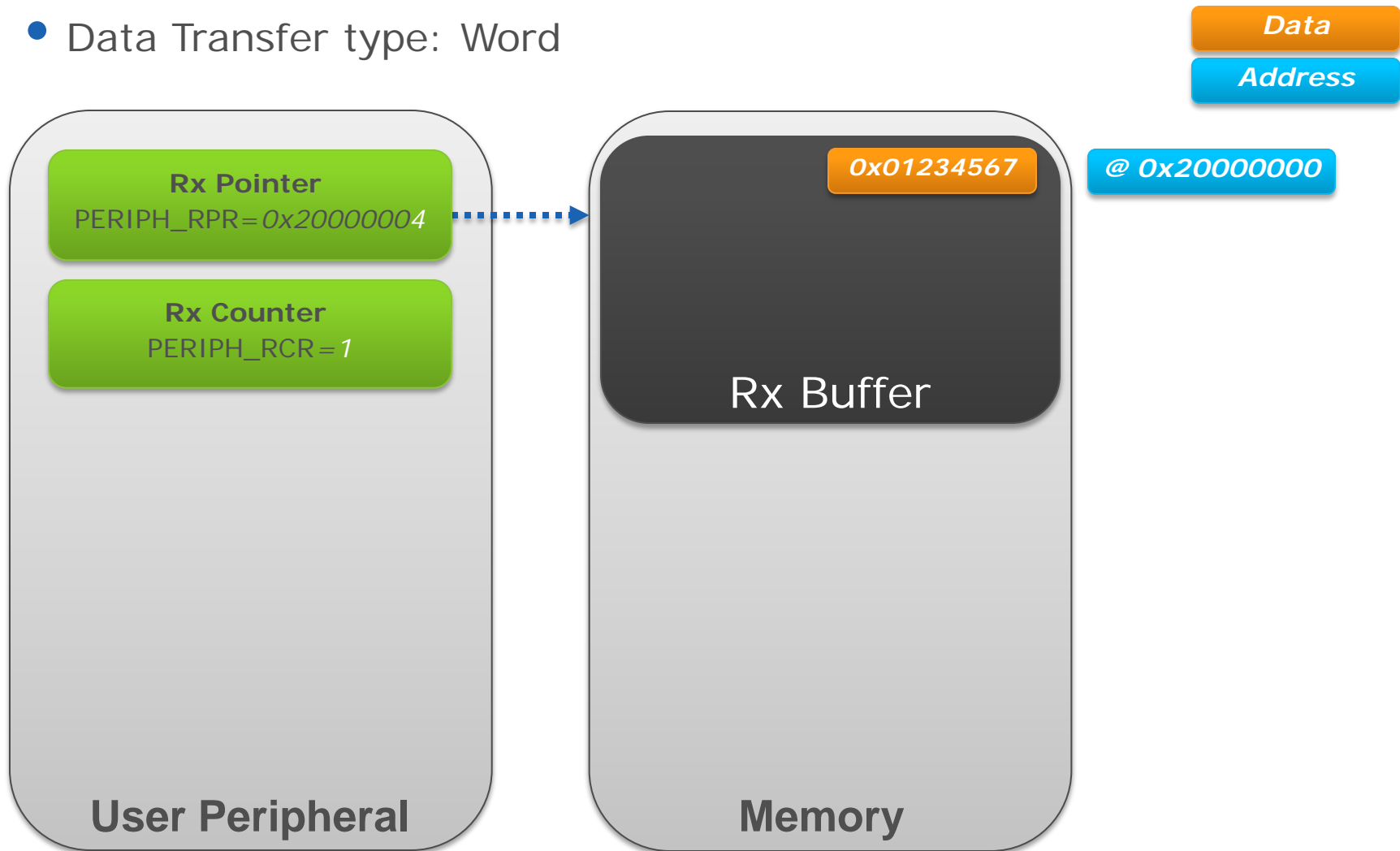
- Data Transfer type: Word



Functional Description

Simple PDC Data Transfer

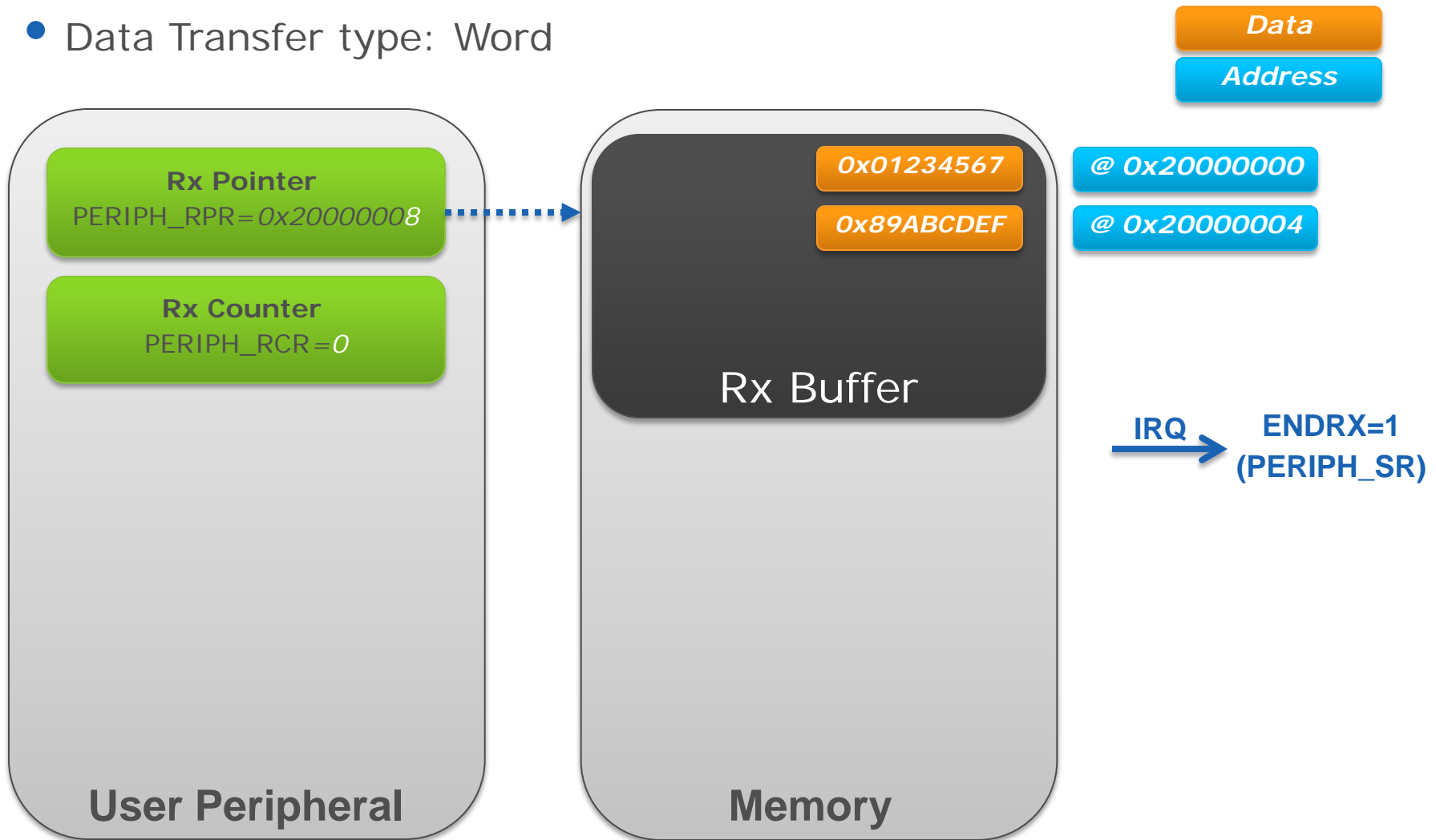
- Data Transfer type: Word



Functional Description

Simple PDC Data Transfer

- Data Transfer type: Word



Functional Description

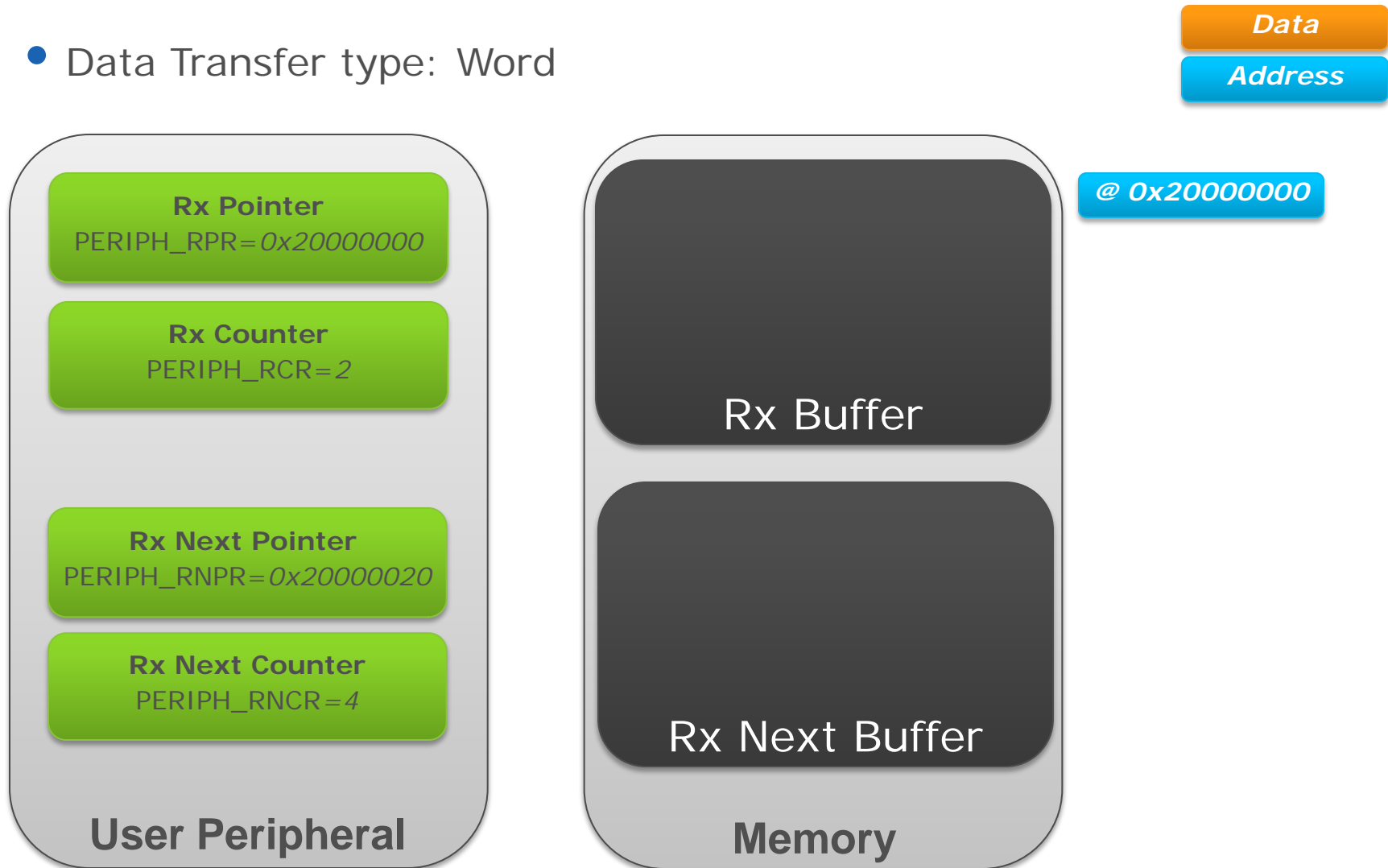
Simple PDC Data Transfer

- How do you program the PDC for 65,535 Tx transfers?
 - Program 0xFFFF in the Transmit Counter Register (TCR)
 - Program Buffer Memory Address in the Transmit Pointer Register (TPR)
 - Enable PDC channel in Transmit (TXTEN)
 - Wait for End Of Transmit Flag (ENDTX=1)
 - Transfer completed

Functional Description

Simple PDC Transfer using Next Pointer/Counter

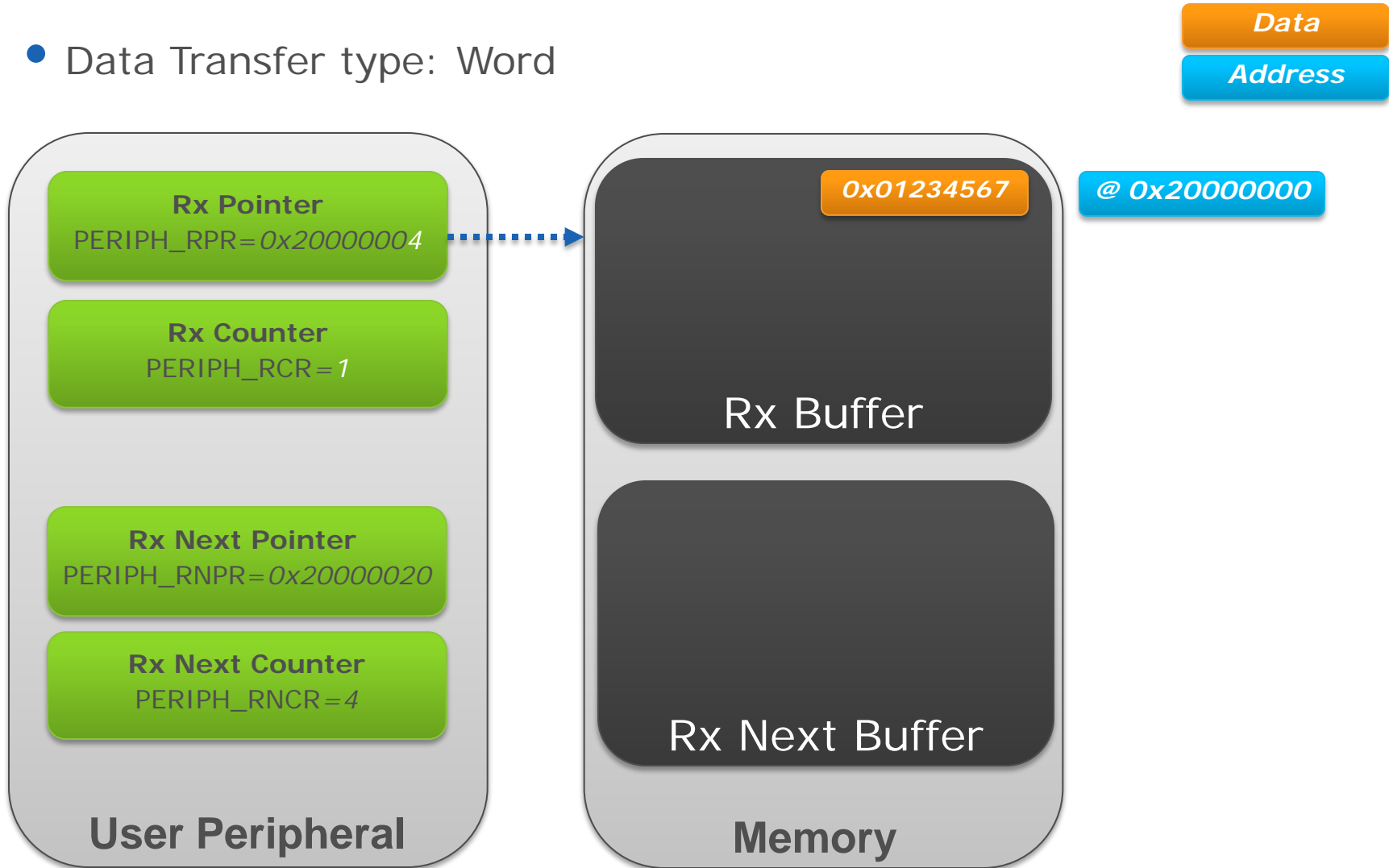
- Data Transfer type: Word



Functional Description

Simple PDC Transfer using Next Pointer/Counter

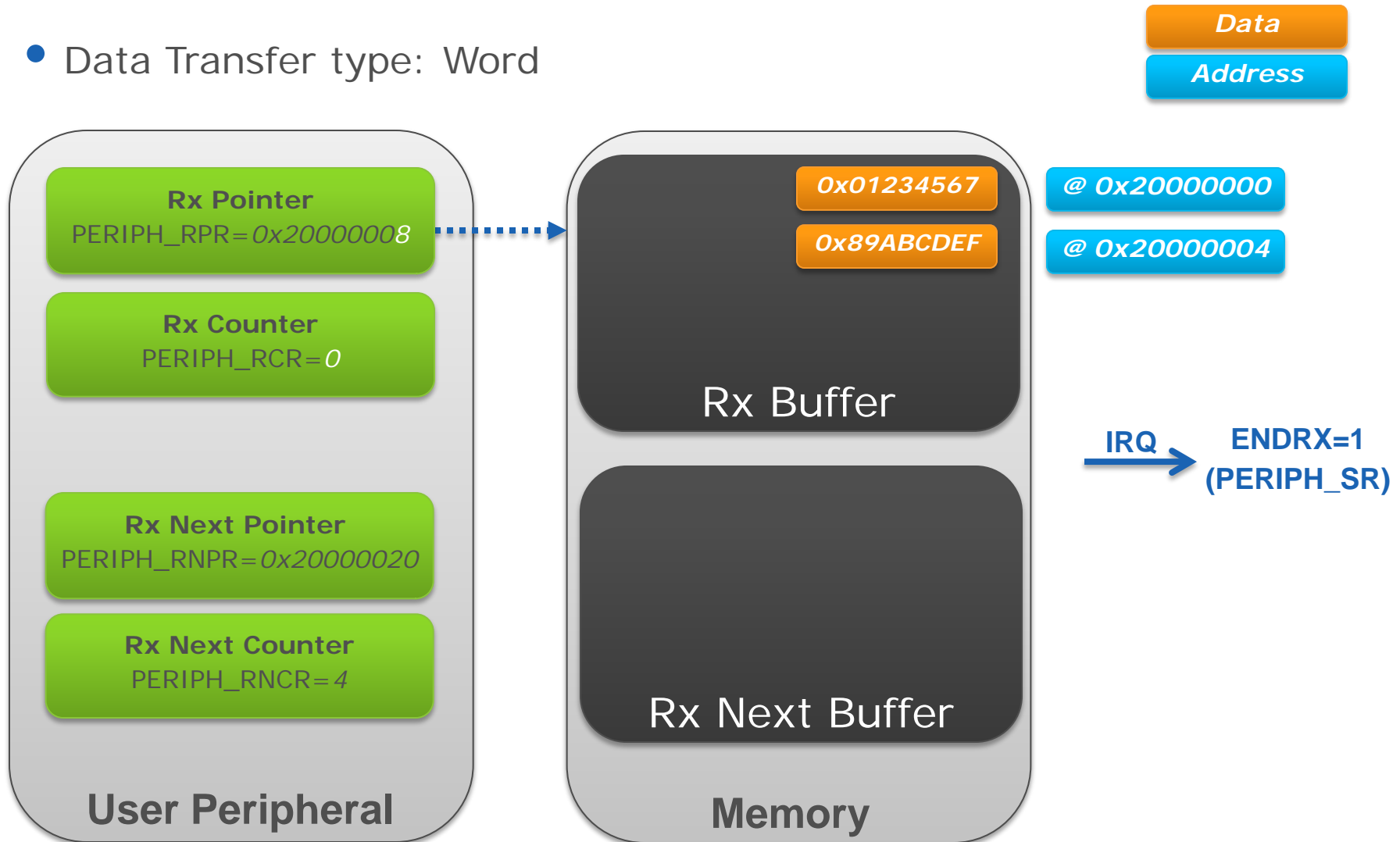
- Data Transfer type: Word



Functional Description

Simple PDC Transfer using Next Pointer/Counter

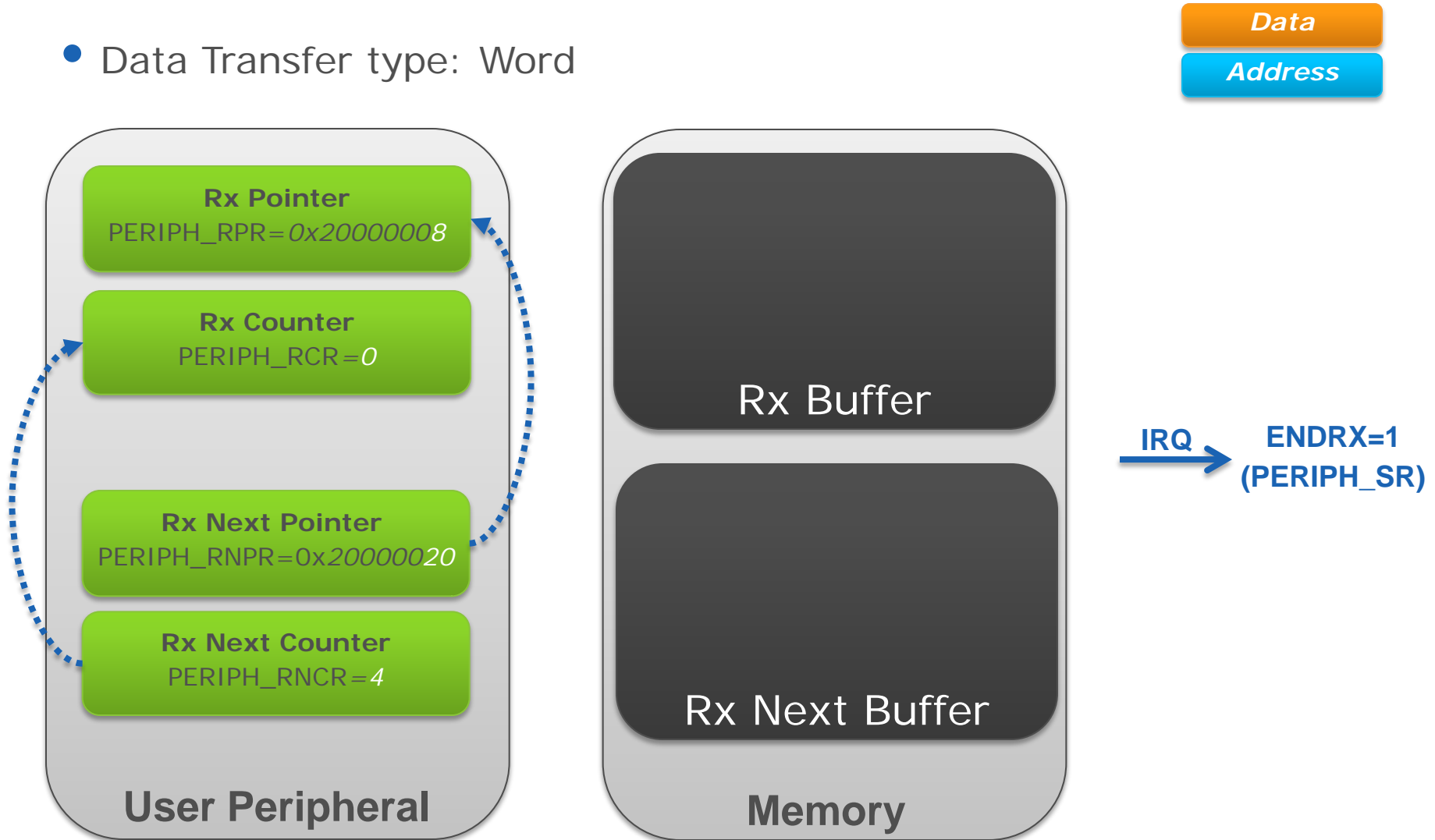
- Data Transfer type: Word



Functional Description

Simple PDC Transfer using Next Pointer/Counter

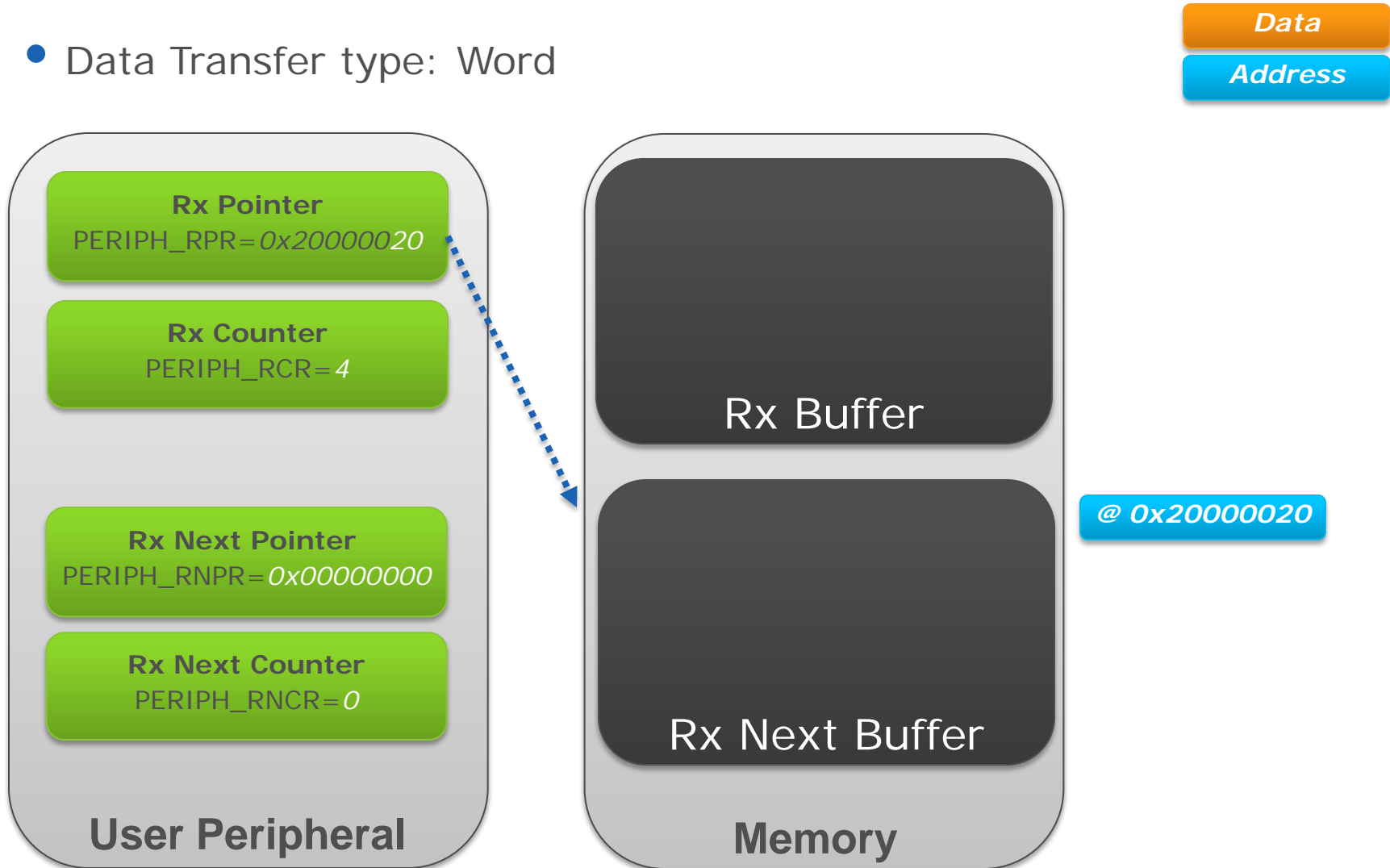
- Data Transfer type: Word



Functional Description

Simple PDC Transfer using Next Pointer/Counter

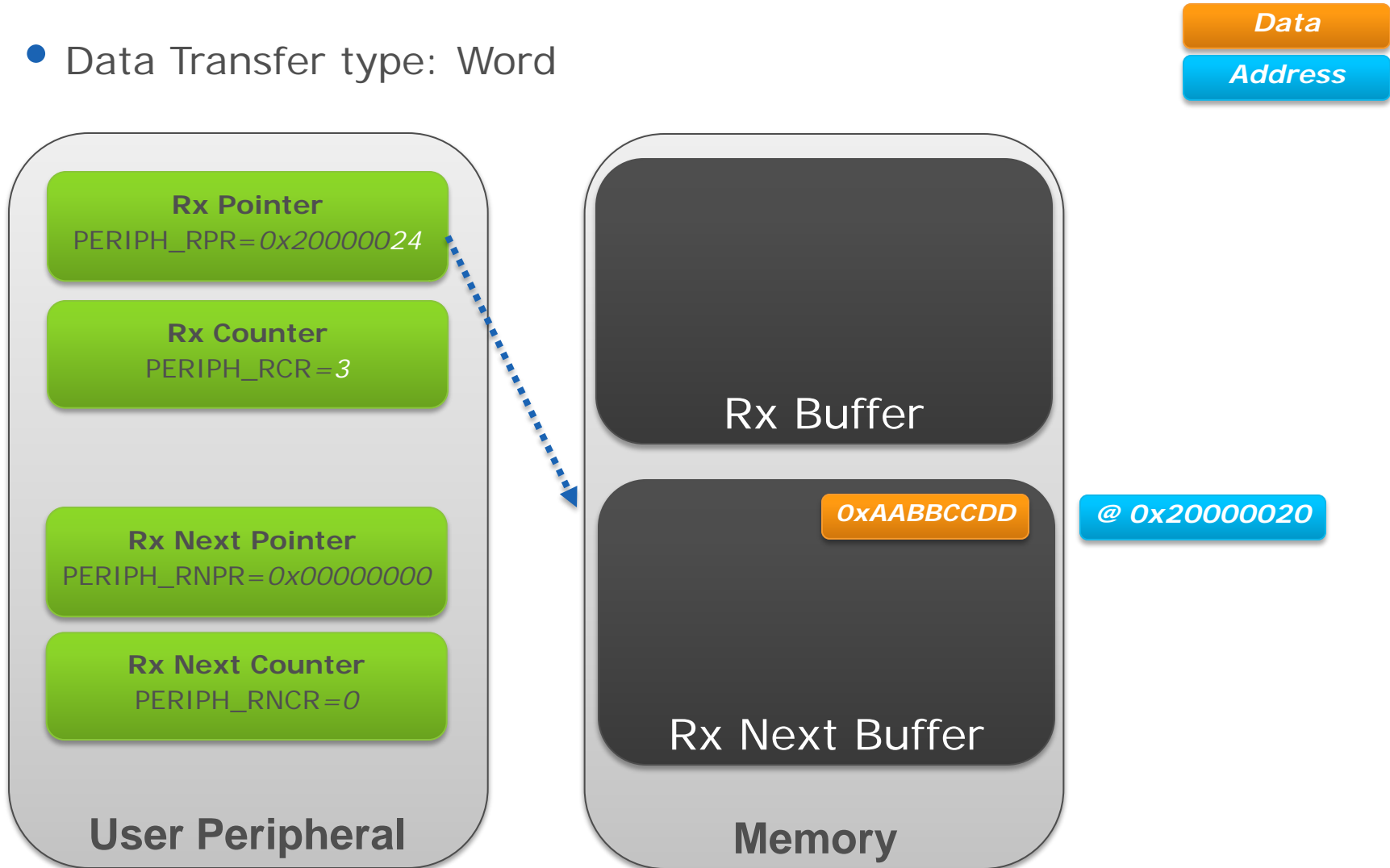
- Data Transfer type: Word



Functional Description

Simple PDC Transfer using Next Pointer/Counter

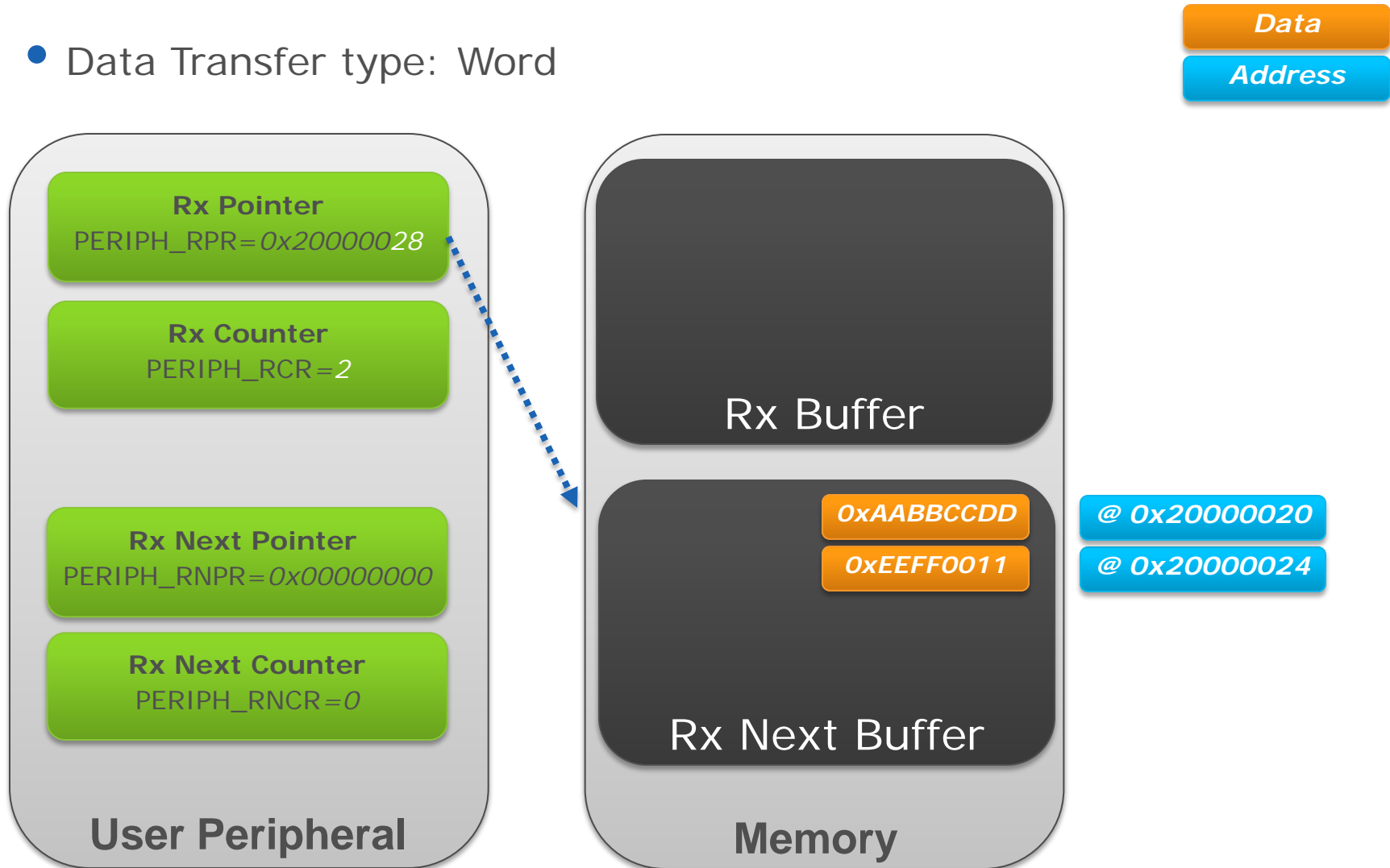
- Data Transfer type: Word



Functional Description

Simple PDC Transfer using Next Pointer/Counter

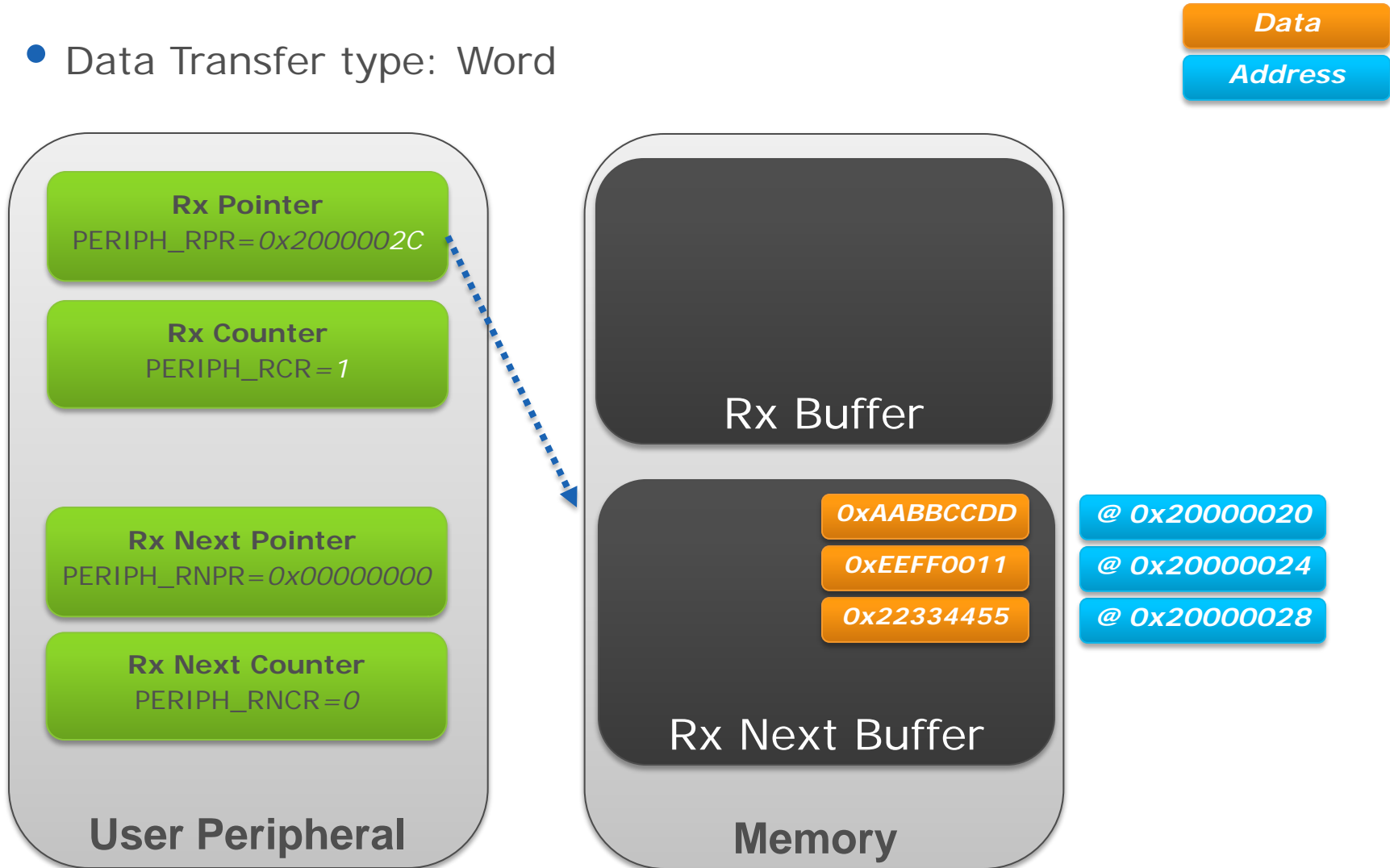
- Data Transfer type: Word



Functional Description

Simple PDC Transfer using Next Pointer/Counter

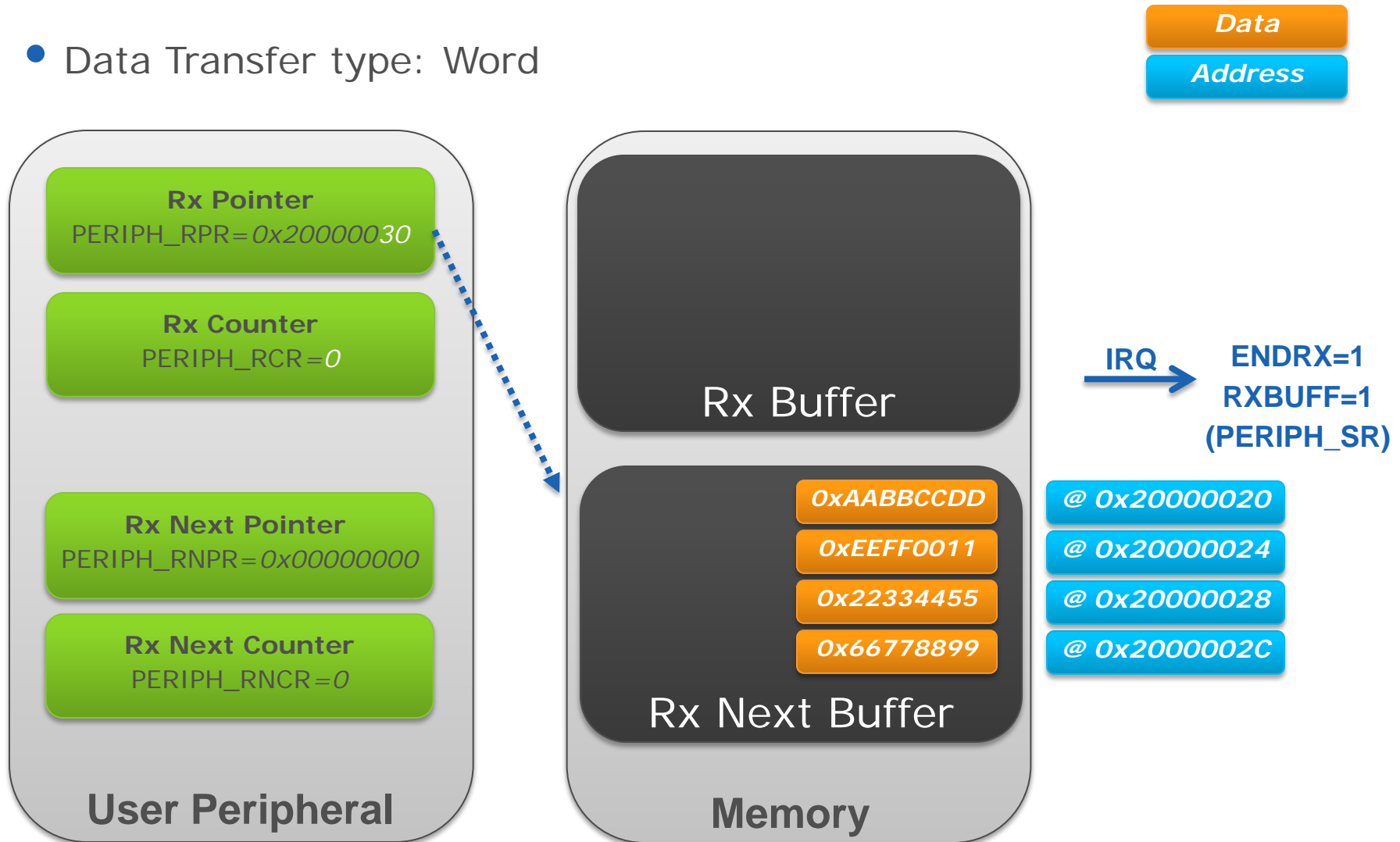
- Data Transfer type: Word



Functional Description

Simple PDC Transfer using Next Pointer/Counter

- Data Transfer type: Word



Functional Description

Simple PDC Transfer using Next Pointer/Counter

- How do you program the PDC for 131,070 Tx transfers?
 - Program 0xFFFF in both TCR & TNCR
 - Program Buffer Memory Addresses in both TPR & TNPR
 - Enable PDC channel in Transmit (TXTEN)
 - Wait for Transmit Buffer Empty (TXBUFE=1)
 - Transfer completed

Functional Description

Continuous PDC Transfer (Buffer Chaining)

- How do you program the PDC for continuous Rx transfers?
 - Program both Counter Registers (RCR & RNCR)
 - Program both Buffer Memory Addresses (RPR & RNPR)
 - Enable PDC channel in Receive (RXTEN)
 - Wait for ENDRX = 1
 - Load RNPR with the next address
 - Load RNCR with the next value
 - If RXBUFF = 1 (i.e. RCR = RNCR = 0)
 - Application Overrun
 - System Arbitration Priority to reconsider

Functional Description

PDC Channels and Data Transfer Type

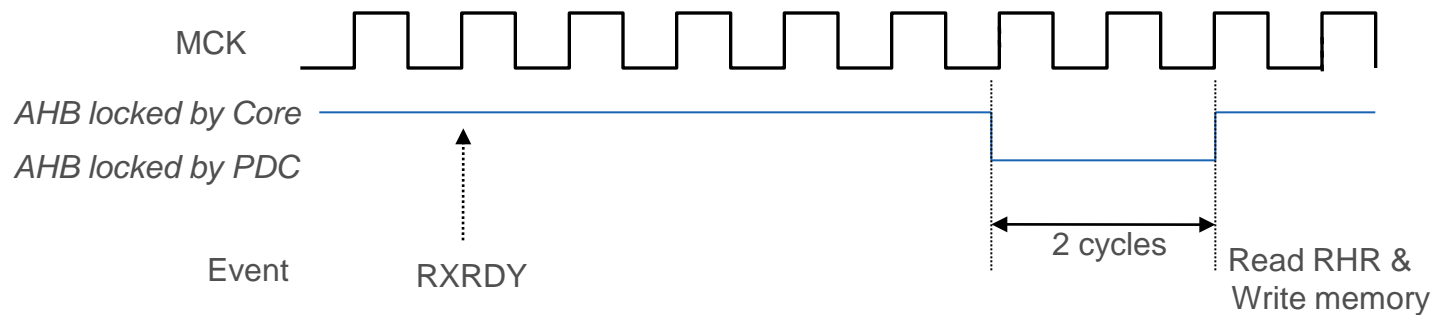
- Depending on Peripheral mode, Data Transfer Type is automatically configured by the PDC
- Example: ADC configured in 10-bit resolution
 - PDC will automatically perform 16-bit data size transfers (MSB filled with '0')

Peripheral	PDC Channels Nb (per Peripheral)	Data Transfer Type	Communication
UART	2	Byte	Full Duplex
USART	2	Byte / Half Word	Full Duplex
SSC	2	Byte / Half Word / Word	Full Duplex
SPI	2	Byte / Half Word / Word	Full Duplex
TWI	2	Byte	Full Duplex
HSMCI	2	Byte / Half Word / Word	Half Duplex
PIOC	1	Byte	Receive Only
ADC	1	Byte / Half Word	Receive Only
DAC	1	Byte / Half Word	Transmit Only
PWM	1	Half Word	Transmit Only

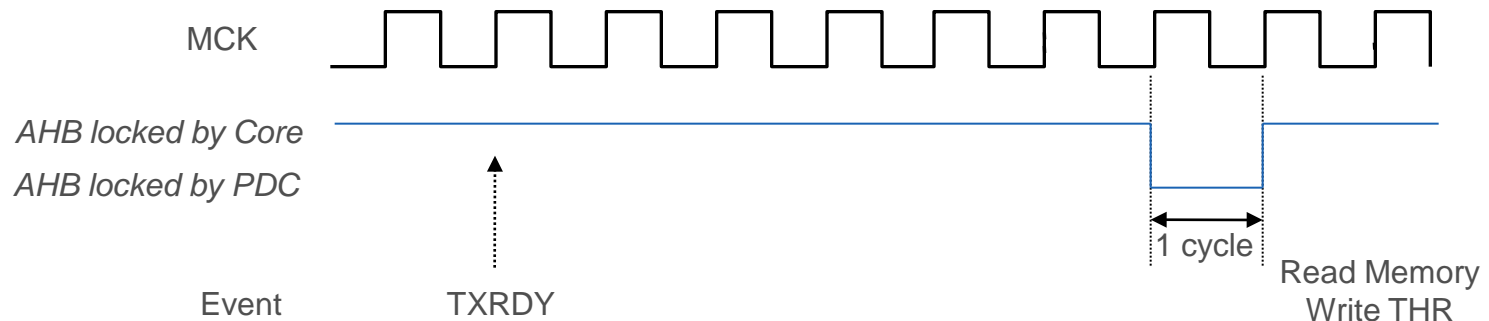
Functional Description

Transfer Delays with AHB bus

- From Peripheral to memory (PDC Receive channel)
 - 2 core cycles



- From Memory to Peripheral (PDC Transmit channel)
 - 1 core cycle

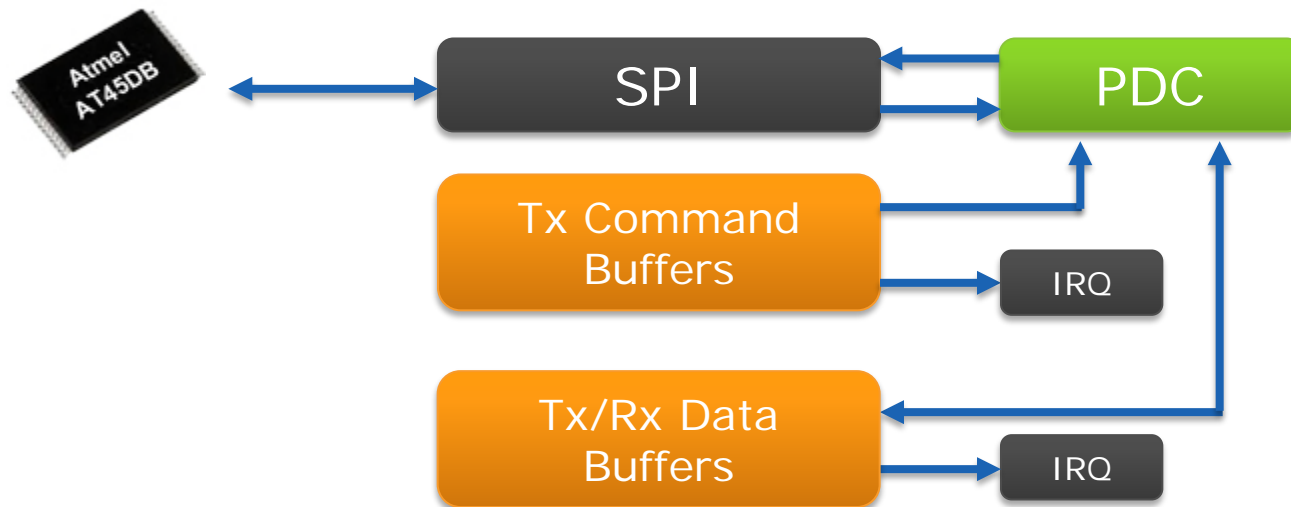


Application Examples

Application Examples

Atmel SPI DataFlash

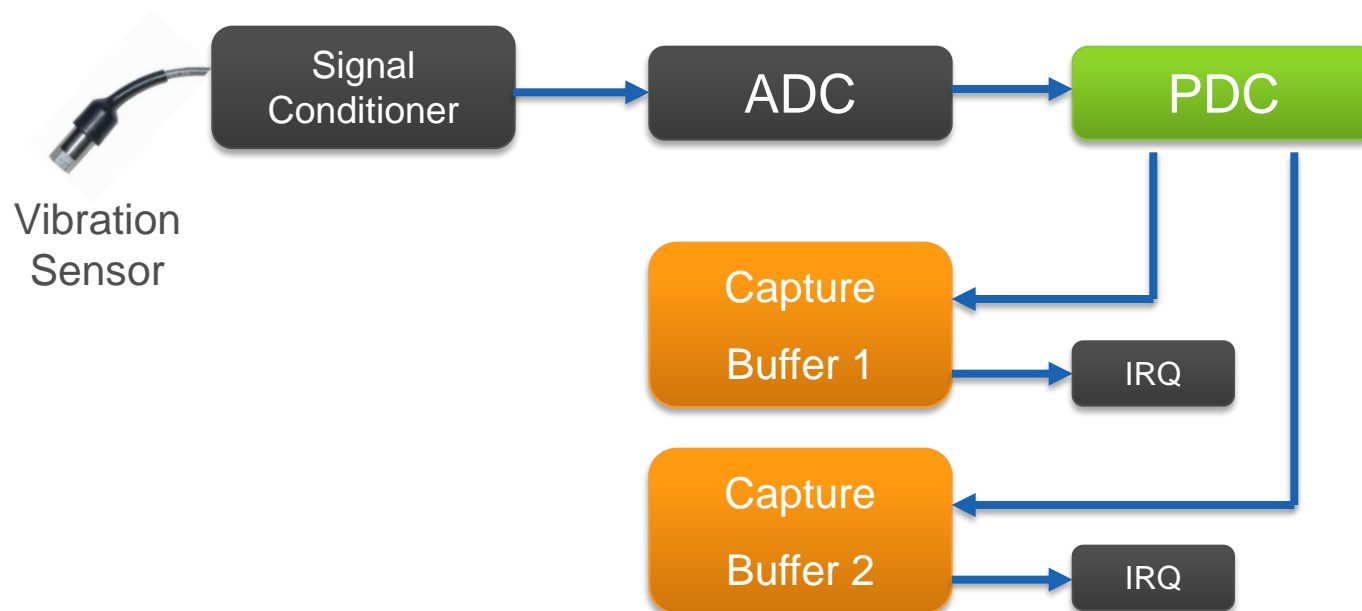
- Transmit Buffers used for Dataflash commands transfer (Read Status, PageWrite, PageRead,...)
- Next Transmit/Receive Buffers used for Dataflash Data transfer



Application Examples

ADC Data Acquisition

- Both Rx buffers (using Next Pointer/Counter registers) are used for data acquisition and Digital Filtering
- When one is used to get ADC samples...
- ... the other one is used by the ARM core to treat the data (Digital Filtering, FFT,...)





Enabling Unlimited Possibilities®

© 2012 Atmel Corporation. All rights reserved.

Atmel®, Atmel logo and combinations thereof, Enabling Unlimited Possibilities®, and others are registered trademarks or trademarks of Atmel Corporation or its subsidiaries. Other terms and product names may be trademarks of others.

Disclaimer: The information in this document is provided in connection with Atmel products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Atmel products. EXCEPT AS SET FORTH IN THE ATMEL TERMS AND CONDITIONS OF SALES LOCATED ON THE ATMEL WEBSITE, ATMEL ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL ATMEL BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS AND PROFITS, BUSINESS INTERRUPTION, OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF ATMEL HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Atmel makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and products descriptions at any time without notice. Atmel does not make any commitment to update the information contained herein. Unless specifically provided otherwise, Atmel products are not suitable for, and shall not be used in, automotive applications. Atmel products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life.