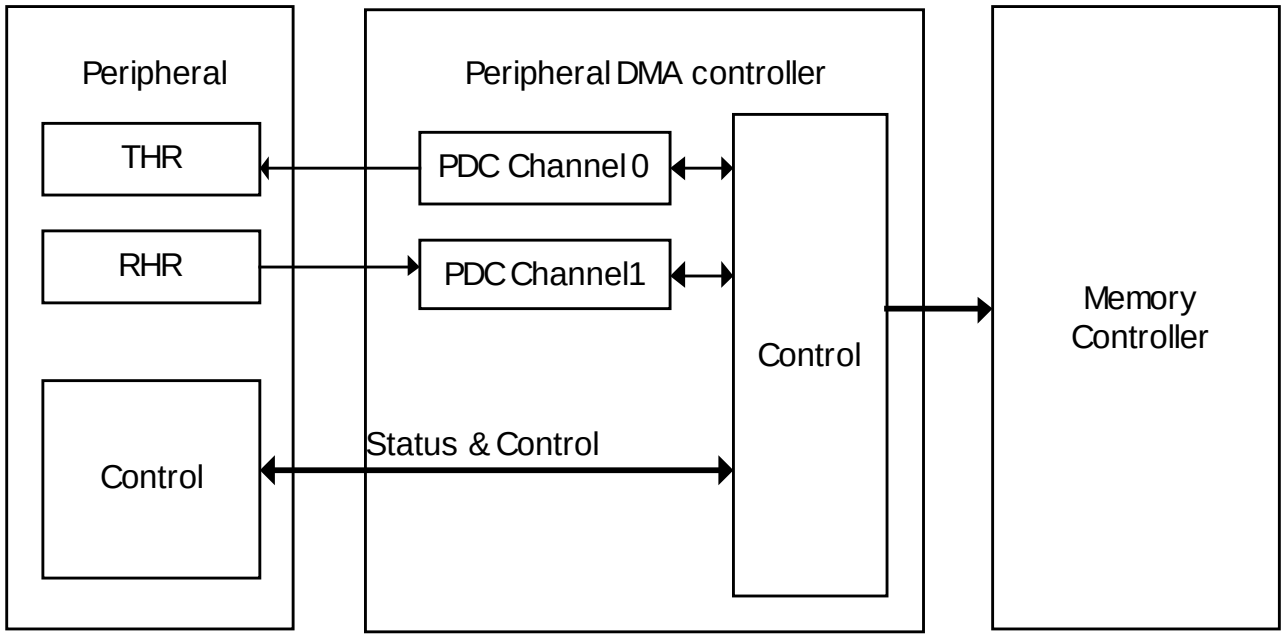
Peripheral DMA Controller (PDC)

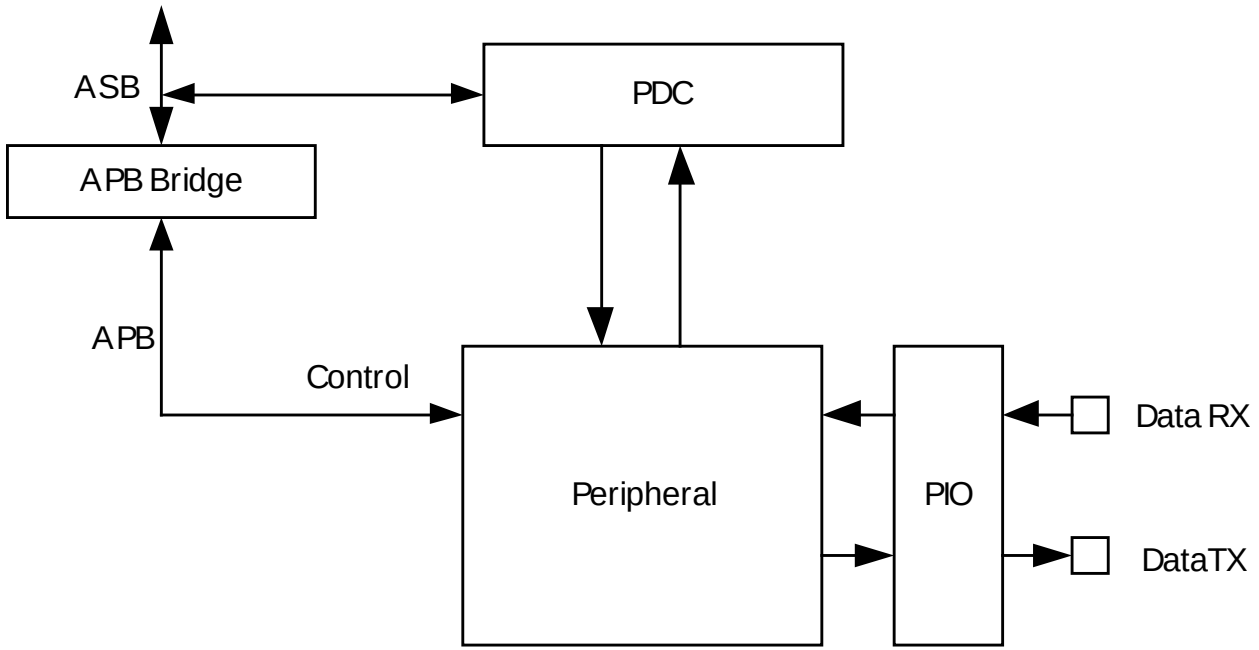
PDC 特征

- 11路 外设 DMA 控制器通道
- 用户接口位于相关外设的用户接口之中
- 无需CPU 支持
  - PDC 传输在 IDLE 模式下仍然可用
- 可编程:
  - 8, 16 与 32-bit 数据尺寸
  - 32-bit 指针寄存器
  - 16-bit 计数寄存器 (64KB 缓冲大小)

PDC 组成框图


Peripheral
THR
RHR
Control
Peripheral DMA controller
PDC Channel 0
PDC Channel1
Control
Status & Control
Memory
Controller

PDC 应用框图


ASB
PDC
APB Bridge
APB
Control
Peripheral
PIO
Data RX
DataTX

依赖关系

- 所有寄存器定义域外设接口
  - DBGU (Rx and Tx)
  - USART 0 (Rx and Tx)
  - USART 1 (Rx and Tx)
  - SSC (Rx and Tx)
  - ADC (Rx)
  - SPI (Rx and Tx)
- PDC 整合于外设的用户接口寄存器偏移量 0x100 处
  - USART0 基地址: 0xFFFC0000
  - USART0 PDC 寄存器起始于 0xFFFC0100

PDC 控制与状态

- 控制
  - 使能
    - 写 1 到 “EN” 位 使能相应通道,只要此时 “DIS” 位未被置位
  - 禁止
    - 写 1 到 “DIS” 位禁止相应通道

PERIPH_PTCR

TXTDIS 9	TXTEN 8
RXTDIS 1	RXTEN 0

- 状态
  - 1 = 对应通道的传输被使能

PERIPH_PTSR

TXTEN 8
RXTEN 0

PDC 寄存器位置

- 控制寄存器位于外设寄存器偏移 0x100 处
  - 0x100 : PERIPH_RPR : Peripheral Receive Pointer Register, Read/Write
  - 0x104 : PERIPH _RCR : Peripheral Receive Counter Register, Read/Write
  - 0x108 : PERIPH _TPR : Peripheral Transmit Pointer Register, Read/Write
  - 0x10C : PERIPH _TCR : Peripheral Transmit Counter Register, Read/Write
  - 0x110 : PERIPH _RNPR : Peripheral Receive Next Pointer Register, Read/Write
  - 0x114 : PERIPH _RNCR : Peripheral Receive Next Counter Register, Read/Write
  - 0x118 : PERIPH _TNPR : Peripheral Transmit Next Pointer Register, Read/Write
  - 0x11C : PERIPH _TNCR : Peripheral Transmit Next Counter Register, Read/Write
  - 0x120 : PERIPH _PTCR : Peripheral PDC Transfer Control Register, Write-only
  - 0x124 : PERIPH _PTSR : Peripheral PDC Transfer Status Register, Read-only

外设 PDC 标志

- 对于每个通道, 两个标志位指明当前缓冲的结束 (ENDRX,ENDTX) 与下一缓冲的结束 (RXBUFF, TXBUFE)
  - ENDRX 位设置于 RCR = 0
  - RXBUFF 标志设置于 RCR = 0 & RNCR = 0
  - ENDTX 设置于 TCR = 0
  - TXBUFE 设置于 TCR = 0 & TNCR = 0

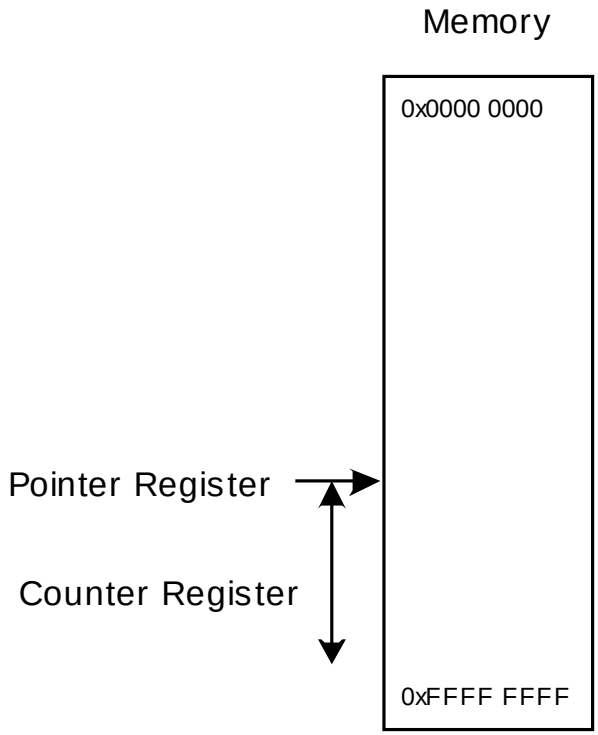
- 这些状态位被直接映射到外设的状态寄存器

外设 PDC 传输

- 每次传输下列的硬件操作被执行
  - 外设通过RXRDY或TXRDY触发 PDC 传输
  - 读取数据 (外设或内存)
  - 写数据 (内存或外设)
  - 增加指针寄存器 (与数据大小 8 ,16 or 32 bits相对应)
  - 计数寄存器减一
  - 检查计数器并设置状态位

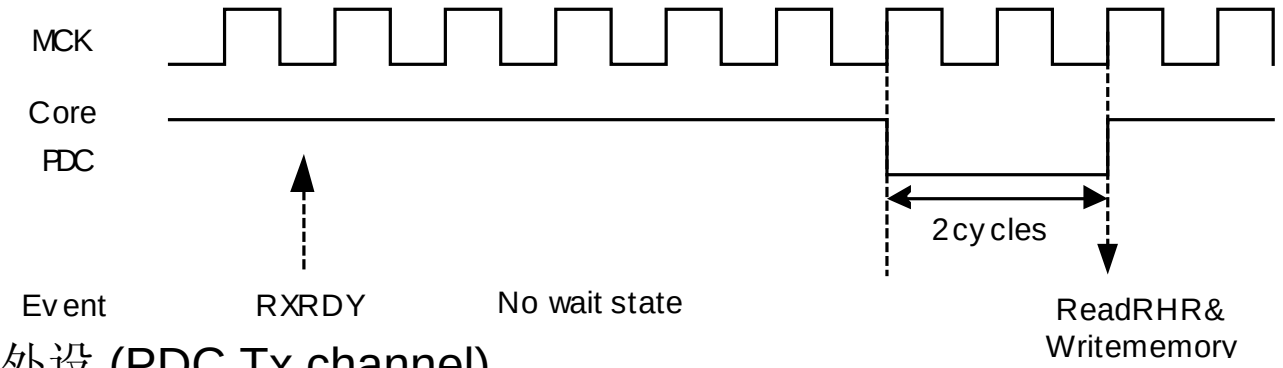
PDC 指针与寄存器

- 数据大小被外设自动处理
  - ADC 8 位采样值被作为8位处理
  - ADC 10 位采样值被作为16位处理,高位补0
- 每个8位 PDC 传输结束之后
  - Pointer = Pointer +1
  - Counter = Counter -1
- 8到16位传输结束之后
  - Pointer = Pointer +2
  - Counter = Counter -1
- 16到32位传输结束之后
  - Pointer = Pointer +4
  - Counter = Counter –1

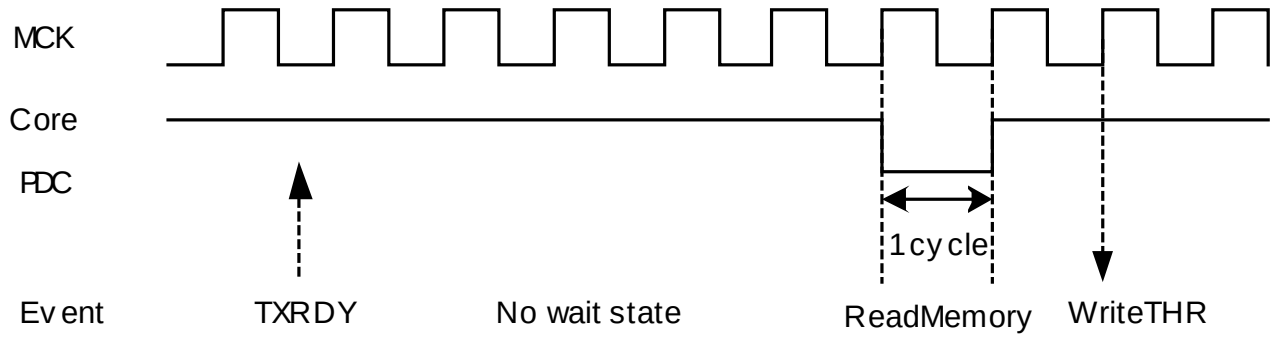


传输延时

- 从外设到内存 (PDC Rx channel)
  - 2 个内核周期



- 从内存到外设 (PDC Tx channel)
  - 1 个内核周期



简单 PDC 传输

- 发送通道例子
  - 初始化外设包括 PDC 初始化寄存器为0
  - 当需要传输时
    - 设置 PDC 指针寄存器 PERIPH_TPR 为内存缓冲地址
    - 设置 PDC 计数寄存器 PERIPH_TCR 为传输大小
    - 清除 下一寄存器 PERIPH_TNPR 与 PERIPH_TNCR
  - 每一次传输都将需求数据写入到保持寄存器
  - 传输完成后置标志位
    - ENDTX 被置位于 TCR = 0 ; TXBUFE 被置位于 TCR & TNCR = 0

简单 PDC 传输

- 如何编程 PDC 进行65,535 次传输?
  - 编程外设
  - 加载 0xFFFF 到 TCR 且设置 TNCR 为 0
  - 加载内存指针 TPR
  - 使能 PDC 通道
  - 当 ENDTX -> 1
    - 传输完成

- 如何编程 PDC 进行 131,070 次传输?
  - 加载 0xFFFF 到 RCR 与 RNCR
  - 加载内存指针 RPR 与 RNPR
  - 使能 PDC 通道
  - 当 RXBUFF -> 1
    - 传输完成

简单 PDC 传输

- 如何编程 PDC 进行穿过 131,070 次的传输?
  - 编程外设, 包括 ENDTX & TXBUFE 中断
  - 加载 0xFFFF 到 TCR & TNCR
  - 加载内存指针 TPR &TNPR
  - 使能 PDC 通道
  - 当 ENDTX -> 1
    - 产生中断
    - TCR = 0
    - 使用下一地址加载 TNPR
    - 用另一个值加载 TNCR
  - 当 TXBUFE -> 1
    - 产生中断
    - TCR=TNCR=0
    - 传输完成

持续 PDC 传输

- 如何编程 PDC 进行持续传输?
  - 编程外设, 包括 ENDRX & RXBUFF 中断
  - 加载相同的值到 RCR & RNCR
  - 加载内存指针 RPR 为起始缓冲地址 而 RNPR 为下一缓冲地址
  - 使能 PDC 通道
  - 当 ENDRX -> 1
    - 产生中断
    - 加载 RNPR 为下一地址
    - 加载 RNCR 为另一个值
  - 当 RXBUFF
    - RCR = RNCR = 0
    - 溢出

AT91 软件库

- 寄存器访问函数
  - AT91F_PDC_SetNextRx
  - AT91F_PDC_SetNextTx
  - AT91F_PDC_SetRx
  - AT91F_PDC_SetTx
  - AT91F_PDC_EnableTx
  - AT91F_PDC_EnableRx
  - AT91F_PDC_DisableTx
  - AT91F_PDC_DisableRx
  - AT91F_PDC_IsTxEmpty
  - AT91F_PDC_IsNextTxEmpty
  - AT91F_PDC_IsRxEmpty
  - AT91F_PDC_IsNextRxEmpty

AT91 软件库

- 设置函数
  - AT91F_PDC_Open
  - AT91F_PDC_Close
  - AT91F_PDC_SendFrame
  - AT91F_PDC_ReceiveFrame

问题解决

- 虚假中断
  - Ready 中断在 外设和 PDC 使能时被置位
- PDC 不能开始
  - PDC 或外设被禁止

PDC 小结

- 大部分传输外设可以使用
  - DBGU, USART, SPI, SSC, ADC
- DMA 控制器用于每一个外设
- 无需 CPU 支持传输
- 无中断传输方式的额外消耗
- 支持持续数据传输
- 可访问所有内存

Team MCUZONE
www.mcuzone.com

2006.1.25