SAM4S User Peripherals

Presentation Outline

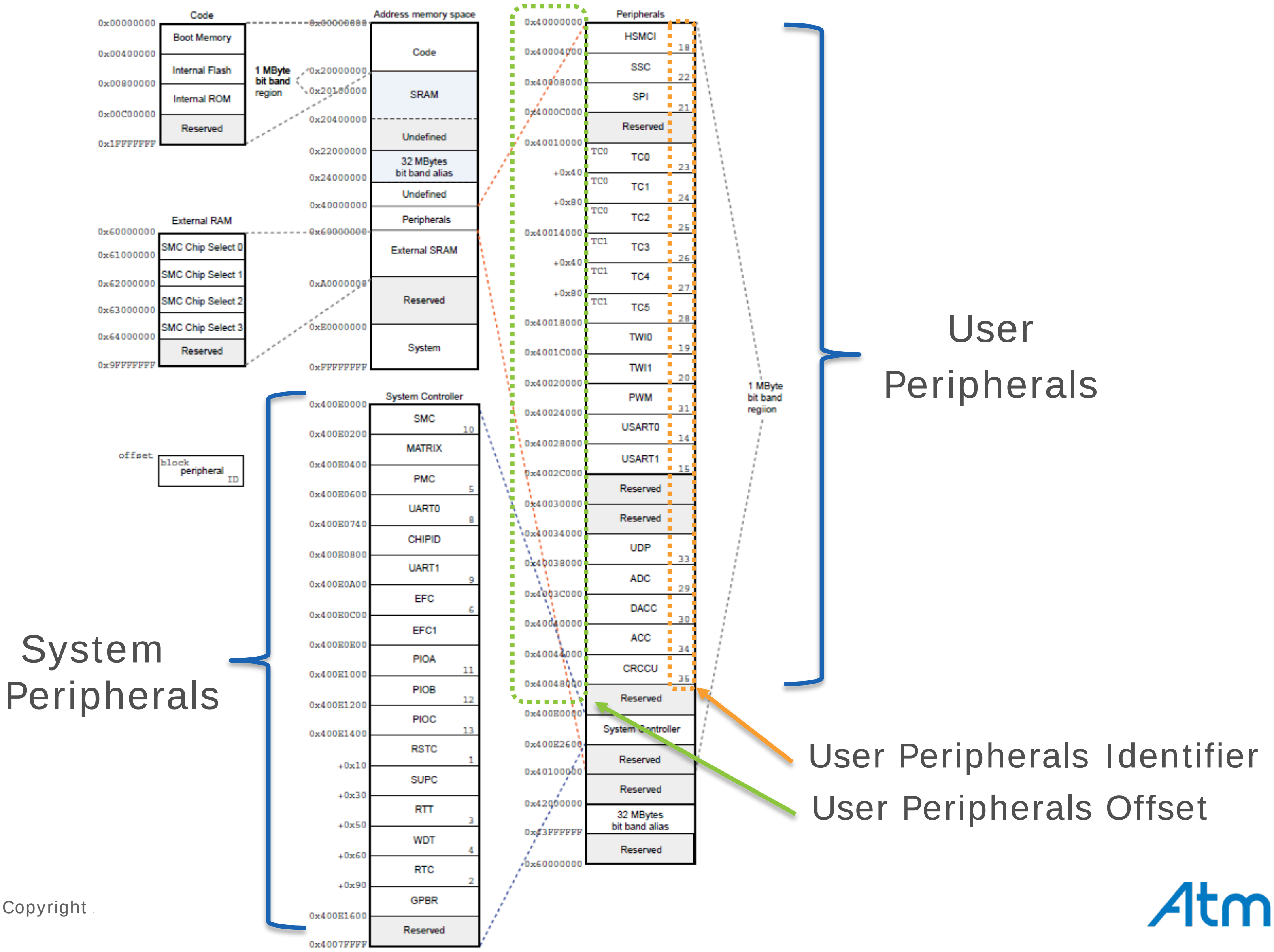
- User Peripherals Initialization
  - Product Mapping
  - Peripheral Identifiers
  - Product Dependencies
  - USART0 Peripheral Initialization Example

- User Peripherals Highlights
  - Serial Peripheral Interface (SPI)
  - Two-wire Interface (TWI)
  - Universal Asynchronous Receiver Transmitter (UART)
  - Universal Synchronous Asynchronous Receiver Transmitter (USART)
  - Synchronous Serial Controller (SSC)
  - High Speed Multimedia Card Interface (HSMCI)
  - Full Speed USB Device Port (UDP)
  - Timer Counter (TC)
  - Pulse Width Modulation Controller (PWM)
  - Analog Comparator Controller (ACC)
  - Analog-to-Digital Converter (ADC)
  - Digital-to-Analog Converter Controller (DACC)
  - CRC Unit (CRCCU)

User Peripherals Initialization

User Peripherals Initialization

Product Mapping



User Peripherals

System Peripherals

User Peripherals Identifier

User Peripherals Offset

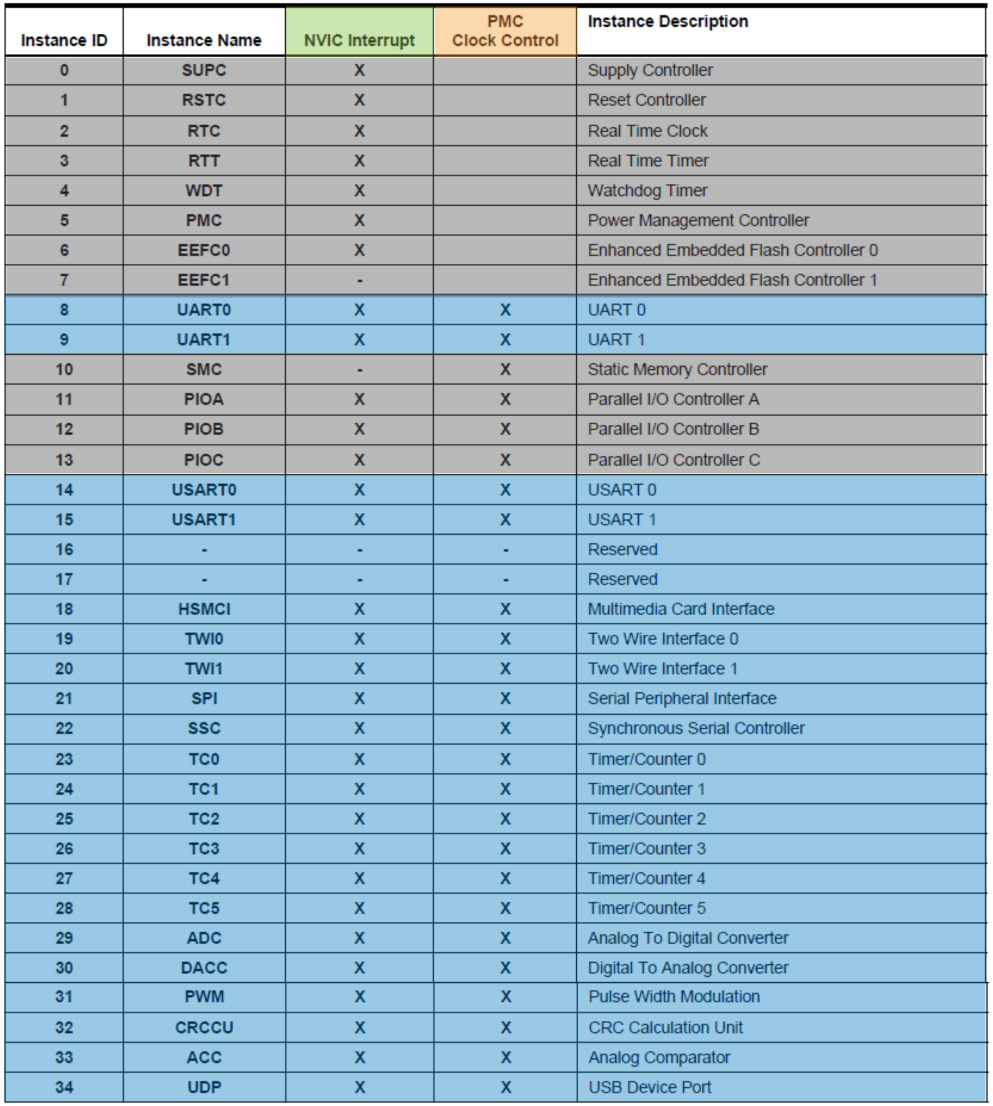
User Peripherals Initialization

Peripheral Identifiers

- A Peripheral Identifier is required for:
  - Peripheral Interrupt Control using the NVIC.
  - Peripheral Clock Control using the PMC.

User Peripherals ID

System Peripherals ID



Instance ID	Instance Name	NVIC Interrupt	PMC Clock Control	Instance Description
0	SUPC	X		Supply Controller
1	RSTC	X		Reset Controller
2	RTC	X		Real Time Clock
3	RTT	X		Real Time Timer
4	WDT	X		Watchdog Timer
5	PMC	X		Power Management Controller
6	EEFC0	X		Enhanced Embedded Flash Controller 0
7	EEFC1	-		Enhanced Embedded Flash Controller 1
8	UART0	X	X	UART 0
9	UART1	X	X	UART 1
10	SMC	-	X	Static Memory Controller
11	PIOA	X	X	Parallel I/O Controller A
12	PIOB	X	X	Parallel I/O Controller B
13	PIOC	X	X	Parallel I/O Controller C
14	USART0	X	X	USART 0
15	USART1	X	X	USART 1
16	-	-	-	Reserved
17	-	-	-	Reserved
18	HSMCI	X	X	Multimedia Card Interface
19	TWI0	X	X	Two Wire Interface 0
20	TWI1	X	X	Two Wire Interface 1
21	SPI	X	X	Serial Peripheral Interface
22	SSC	X	X	Synchronous Serial Controller
23	TC0	X	X	Timer/Counter 0
24	TC1	X	X	Timer/Counter 1
25	TC2	X	X	Timer/Counter 2
26	TC3	X	X	Timer/Counter 3
27	TC4	X	X	Timer/Counter 4
28	TC5	X	X	Timer/Counter 5
29	ADC	X	X	Analog To Digital Converter
30	DACC	X	X	Digital To Analog Converter
31	PWM	X	X	Pulse Width Modulation
32	CRCCU	X	X	CRC Calculation Unit
33	ACC	X	X	Analog Comparator
34	UDP	X	X	USB Device Port

User Peripherals Initialization

Product Dependencies

- **I/O Lines (PIO)**
  - User Peripheral pins may be multiplexed with PIO lines.
  - The programmer must first program the PIO controllers to assign the User Peripheral pins to their peripheral functions.
  - Other I/Os features such as pull-up/down or multi drive may be configured too.

- **Power Management (PMC)**
  - The User Peripheral may be clocked through the Power Management Controller.
  - The programmer must first configure the PMC to enable the User Peripheral clock.

- **Interrupt (NVIC)**
  - The User Peripheral interface has an interrupt line connected to the NVIC Interrupt Controller.
  - Handling the User Peripheral interrupt requires programming the interrupt controller before configuring the User Peripheral.

- **Peripheral DMA (PDC) – (if available for the User Peripheral)**
  - The User Peripheral interface may be used in conjunction with the PDC in order to reduce processor overhead.

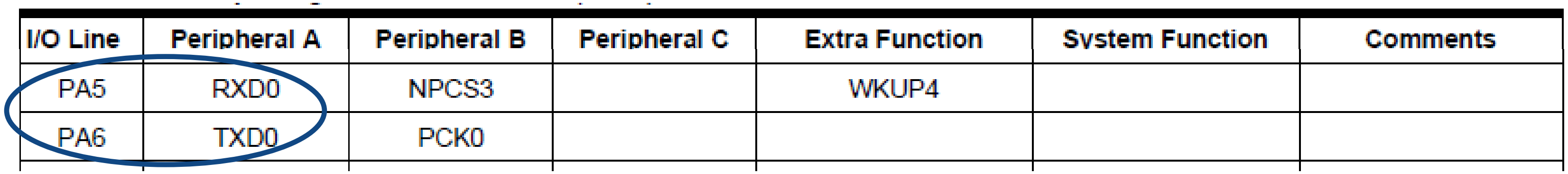
User Peripherals Initialization

USART0 Peripheral Initialization Example

1. I/O Lines (PIO)

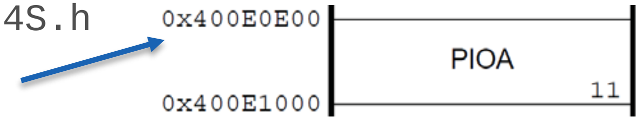
The PIO Controller provides multiplexing of up to four peripheral functions on a single pin.

- The selection is performed by writing PIO_ABCDSR1 / PIO_ABCDSR2 (ABCD Select Registers).



I/O Line	Peripheral A	Peripheral B	Peripheral C	Extra Function	System Function	Comments
PA5	RXD0	NPCS3		WKUP4		
PA6	TXD0	PCK0				

```
// PIOA Controller Base Address defined in SAM4S.h
#define PIOA        ((Pio     *)0x400E0E00U)
```



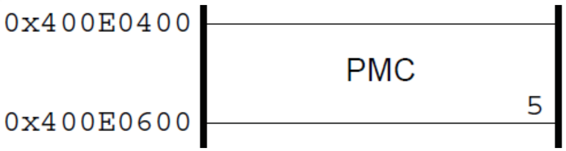
```
// Clear PA5/PA6 bits in PIO_ABCDSR1 to assign I/O lines to Peripheral A
PIOA->PIO_ABCDSR[0] &= ~((1 << 5)|(1 << 6));
// Clear PA5/PA6 bits in PIO_ABCDSR2 to assign I/O lines to Peripheral A
PIOA->PIO_ABCDSR[1] &= ~((1 << 5)|(1 << 6));

// Enable USART0 RXD0 (PA5) & TXD0 (PA6)
PIOA->PIO_PDR = (1 << 5)|(1 << 6);
```

User Peripherals Initialization

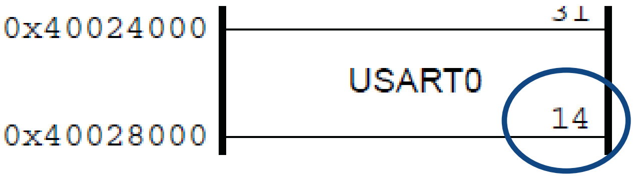
USART0 Peripheral Initialization Example (cont.)

2. Power Management (PMC)

```
// PMC Controller Base Address defined in SAM4S.h
#define PMC ((Pmc    *)0x400E0400U)
```



```
// Enable USART0 PMC Clock
PMC->PMC_PCER1 = (1 << 14);
```

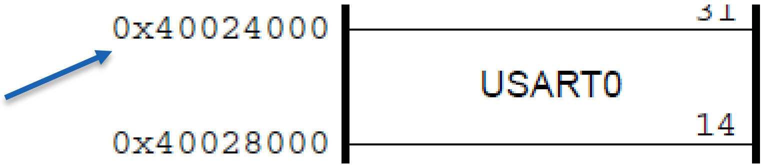


User Peripherals Initialization

USART0 Peripheral Initialization Example (cont.)

3. User Peripheral Configuration
   - Asynchronous
   - 115200 - 8bits – No Parity – 1 Stop bit

```
// USART 0 Base Address defined in SAM4S.h
#define USART0      ((Usart  *)0x40024000U
```



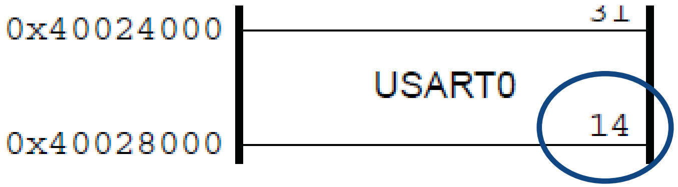
```
// Reset and disable receiver & transmitter
USART0->US_CR= US_CR_RSTRX|US_CR_RSTTX|US_CR_RXDIS|US_CR_TXDIS;

// Configure USART0 Mode Register
USART0->US_MR = US_MR_USART_MODE_NORMAL|US_MR_CHRL_8_BIT| US_MR_PAR_NO|
  US_MR_NBSTOP_1_BIT;

// Configure USART0 Baudrate
USART0->US_BRGR = MCK/115200;

// Enable TX and RX
USART0->US_CR = US_CR_TXEN | US_CR_RXEN;
```

User Peripherals Initialization

USART0 Peripheral Initialization Example (cont.)

4. Interrupt (NVIC)

```
// Enable IRQn (CMSIS function)
NVIC_EnableIRQ(14);
```



5. Peripheral DMA (PDC)

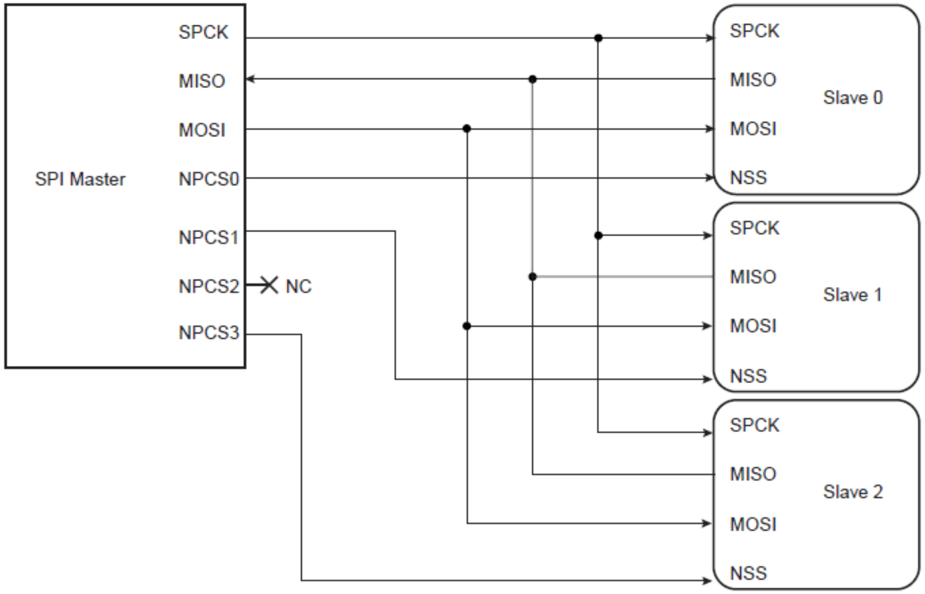
```
// Configure Receive PDC Channel
USART0->US_RPR = (uint32_t) buffer; // buffer pointer
USART0->US_RCR = size; // size of the transfer
USART0->US_PTCR = US_PTCR_RXTEN; // Enable PDC RX Channel

// Enable ENDRX PDC Interrupt
USART0->US_IER = US_IER_ENDRX;

// Wait now for ENDRX flag ⇔ End of Rx Transfer
// Specific Application Code
...
```

User Peripherals Highlights

User Peripherals Highlights

Serial Peripheral Interface (SPI)

- Supports communication with serial external devices
  - 4 hardware chip selects
  - Up to 15 peripherals using external decoder
  - Serial memories (DataFlash...)
  - Serial peripherals (ADC, DAC...)

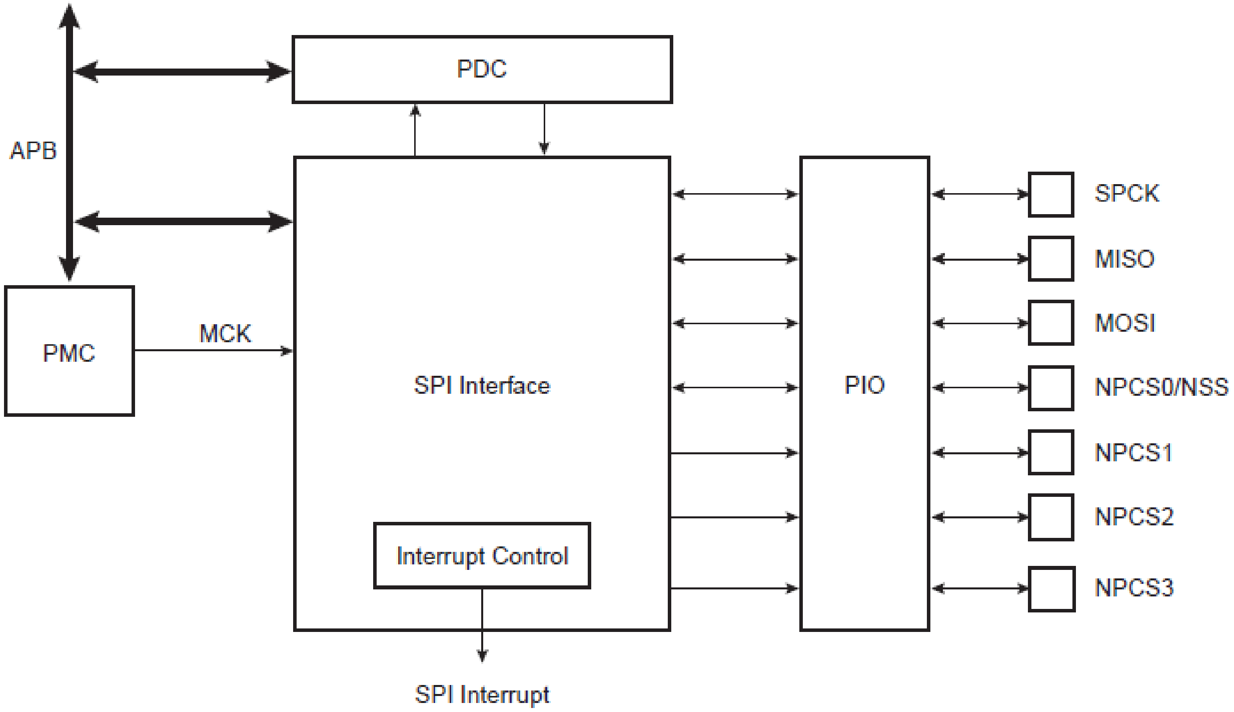


- Master / Slave
  - 8- to 16-bit programmable data length per chip select
  - SPI modes 0, 1, 2 and 3
    - Programmable phase and polarity per chip select
  - Programmable delays between:
    - Chip selects
    - Consecutive transfers
    - Clock and data per chip select

User Peripherals Highlights

Serial Peripheral Interface (SPI)

- Master Mode can run SPI Clock (SPCK) up to MCK

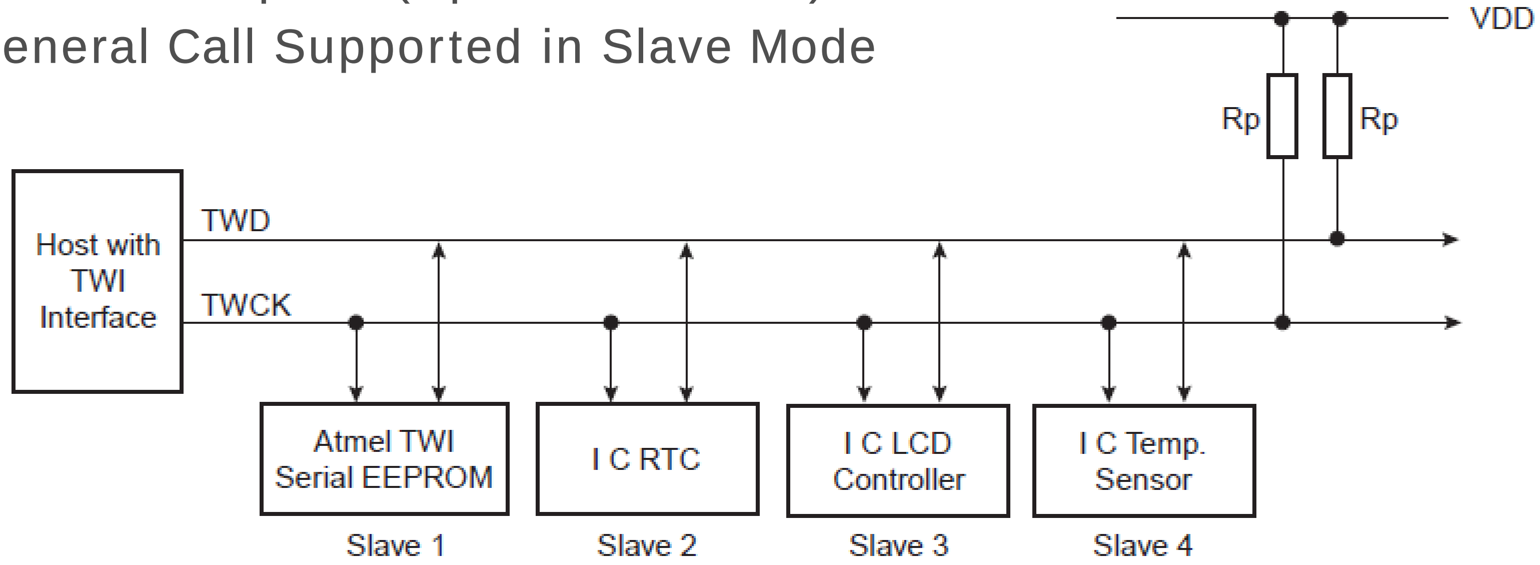


- PDC Support
  - One Transmit PDC channel + One Receive PDC channel

User Peripherals Highlights

Two-Wire Interface (TWI)

- Master, Multi-Master and Slave Mode Operation
- Compatibility with I2C compatible devices
- One, two or three bytes for slave address
- Fast Mode Speed (Up to 400 kbit/s)
- General Call Supported in Slave Mode

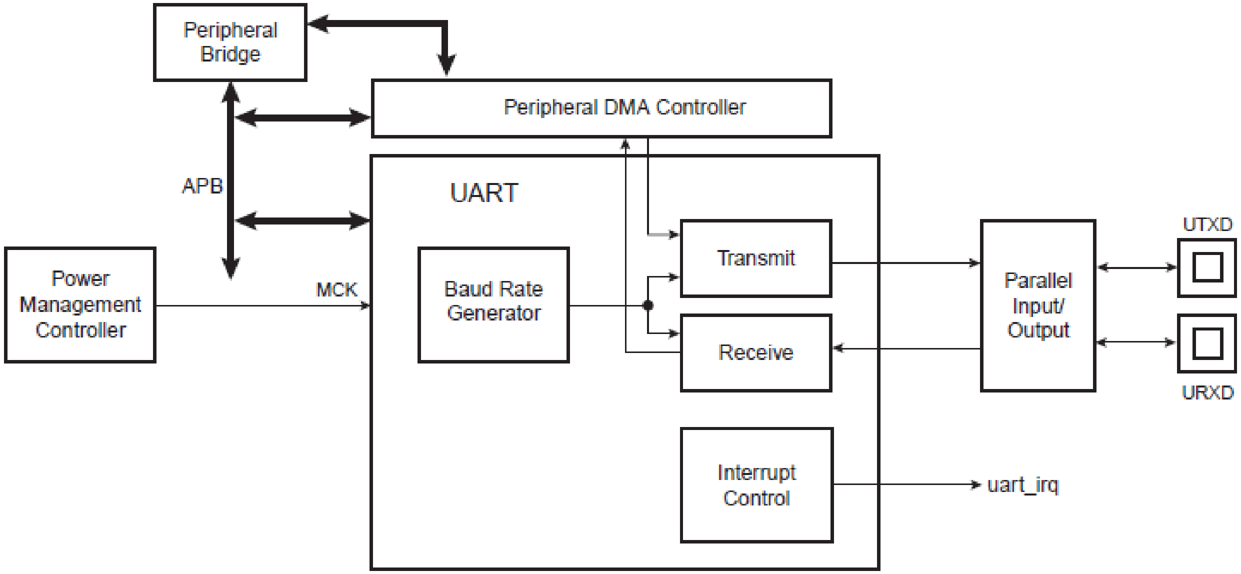


- PDC Support
  - One Transmit PDC channel + One Receive PDC channel

User Peripherals Highlights

Universal Asynchronous Receiver Transmitter (UART)

- Two-pin UART
  - Independent receiver and transmitter with a Baud Rate Generator
  - Even, Odd, Mark or Space Parity Generation
  - Automatic Echo, Local Loopback and Remote Loopback Channel Modes



- PDC Support
  - One Transmit PDC channel + One Receive PDC channel

User Peripherals Highlights

Universal Synchronous Asynchronous Receiver Transmitter (USART)

- 5- to 9-bit full-duplex synchronous/asynchronous serial com
    - Hardware handshaking RTS-CTS
    - Optional Manchester Encoding
    - Full modem line support on USART1 (DCD-DSR-DTR-RI)
- RS485 with driver control signal
- ISO7816, T = 0 or T = 1 Protocols for interfacing with smart cards
- SPI Mode
    - Master or Slave
    - Serial Clock programmable Phase and Polarity
    - SPI Serial Clock (SCK) Frequency up to MCK/4
- IrDA modulation and demodulation
    - Communication at up to 115.2 Kbps

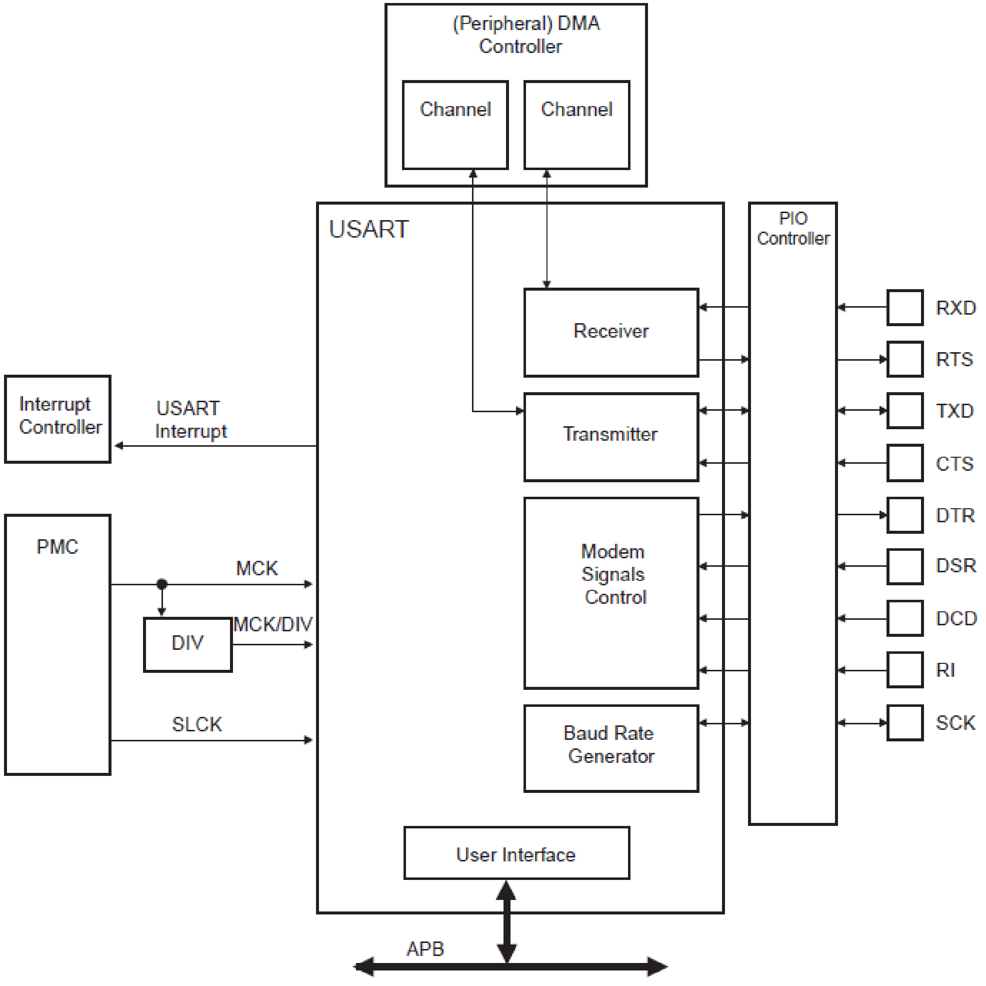
User Peripherals Highlights

Universal Synchronous Asynchronous Receiver Transmitter (USART)

- Common USART Signals:
  - SCK Serial Clock
  - RXD Receive Data
  - RTS Request To send
  - TXD Transmit Data
  - CTS Clear To Send

- Modem Signals:
  - DTR Data Terminal Ready
  - DSR Data Set Ready
  - DCD Data Carrier Detect
  - RI Ring Indicator

- PDC Support
  - One Transmit PDC channel
  - One Receive PDC channel

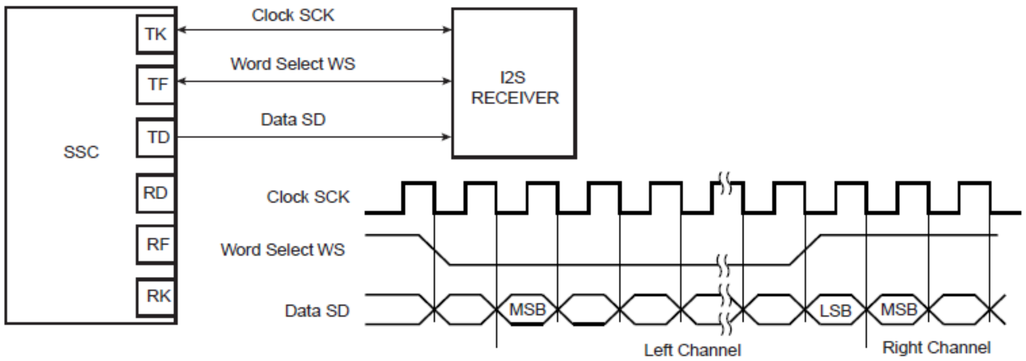


User Peripherals Highlights

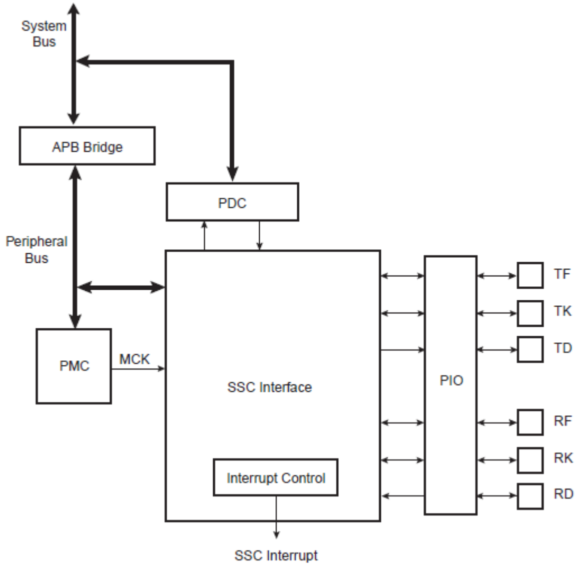
Synchronous Serial Controller (SSC)

- Provides serial synchronous communication links with external devices
  - CODECs, I2S, TDM Buses...

Figure 31-17.Audio Application Block Diagram



- Receiver and transmitter include:
  - a data signal
  - a clock signal
  - a frame synchronization signal

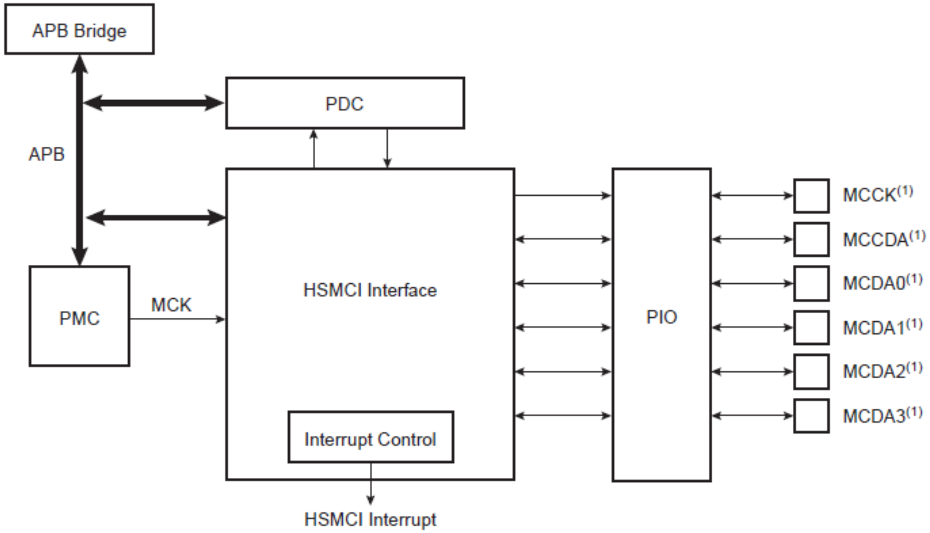


- PDC Support
  - One Transmit PDC channel
  - One Receive PDC channel

User Peripherals Highlights

High Speed MultiMedia Card Interface (HSMCI)

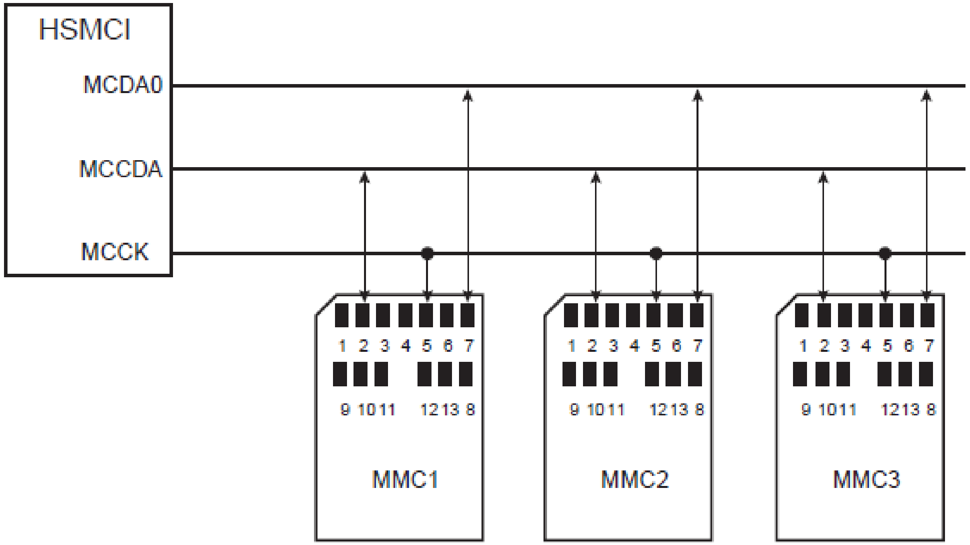
- 1-bit or 4-bit Interface
- Compatibility with:
  - MultiMedia Card Specification Version 4.3
  - SD and SDHC Memory Card Specification Version 2.0
  - SDIO Specification Version V1.1.
- High Speed mode support
- Cards clock rate up to MCK/2
- PDC Support
  - 1 Common PDC channel for transmit/receive (half duplex)



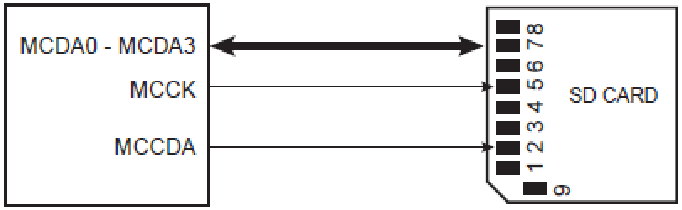
User Peripherals Highlights

High Speed MultiMedia Card Interface (HSMCI)

- MCI has one slot supporting
  - One MultiMediaCard bus (up to 30 cards)



  - Or one SD Memory Card
  - Or one SDIO Card

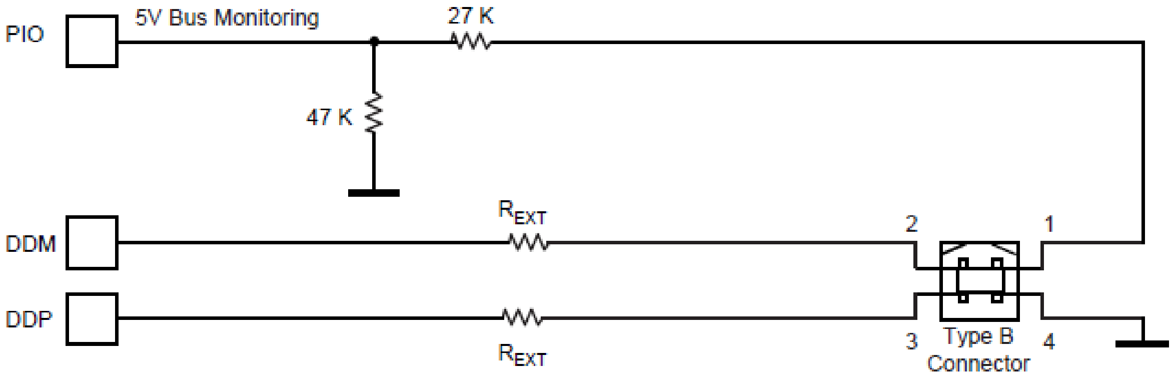


User Peripherals Highlights

USB Device Port (UDP)

- USB V2.0 full-speed compliant, 12 Mbits per second.
  - Eight endpoints
  - Embedded 2688-byte Dual-Port RAM for endpoints
  - Suspend/resume logic

- Embedded USB V2.0 full-speed transceiver



- Integrated Pull-up on DDP
- Pull-down resistors on DDM and DDP when disabled

User Peripherals Highlights

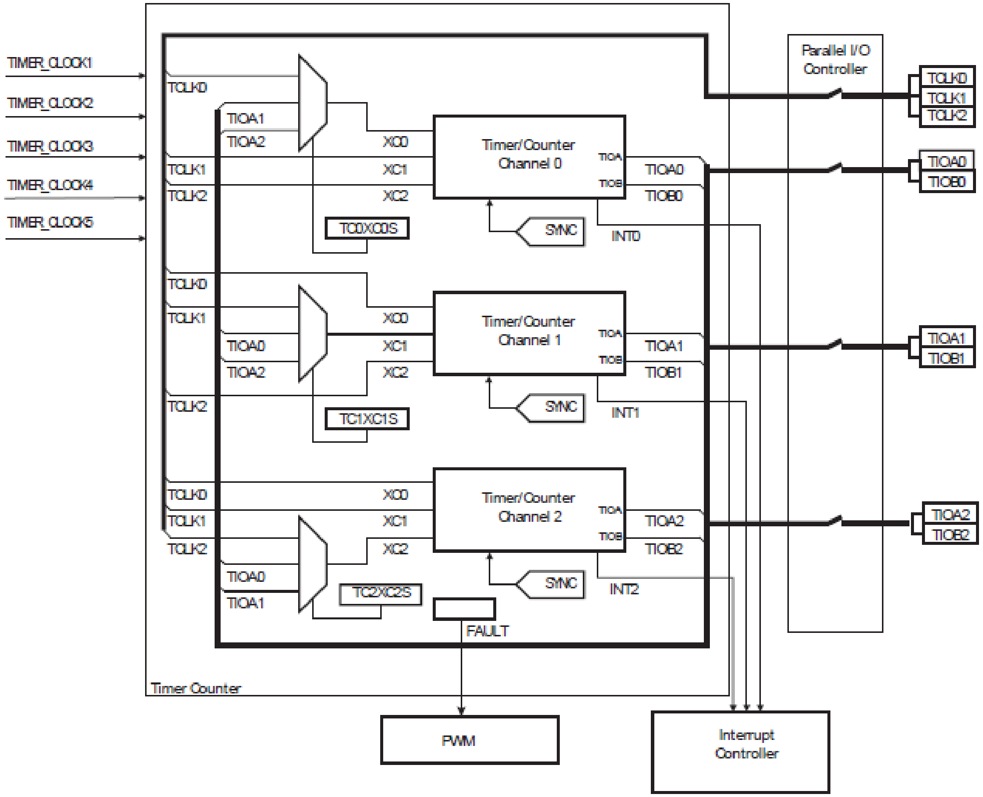
Timer Counter (TC)

- Six 16-bit Timer Counter Channels
  - Two Timer Counters / Three 16-bit TC Channels per Timer Counter
- Wide range of functions including:
  - Frequency Measurement
  - Event Counting
  - Interval Measurement
  - Pulse Generation
  - Delay Timing
  - Pulse Width Modulation
  - Up/down Capabilities
- Quadrature decoder
  - Position / revolution / speed
- 2-bit Gray Up/Down Counter for Stepper Motor

User Peripherals Highlights

Timer Counter (TC)

- Each channel is user-configurable and contains:
  - Three external clock inputs: TCLKx
  - Five internal clock inputs: TIMER_CLOCKx
  - Two multi-purpose input/output signals: TIOA/TIOB



User Peripherals Highlights

Pulse Width Modulation Controller (PWM)

- One Four-channel 16-bit PWM Controller
  - 16-bit counter per channel

- Common clock generator, providing Thirteen Different Clocks
  - A Modulo n counter providing eleven clocks
  - Two independent Linear Dividers working on modulo n counter outputs

- Independent channel programming
  - Independent Enable Disable Commands
  - Independent Clock Selection
  - Independent Period and Duty Cycle, with Double Buffering
  - Programmable selection of the output waveform polarity
  - Programmable center or left aligned output waveform
  - Independent complementary Outputs with 12-bit dead time generator

User Peripherals Highlights

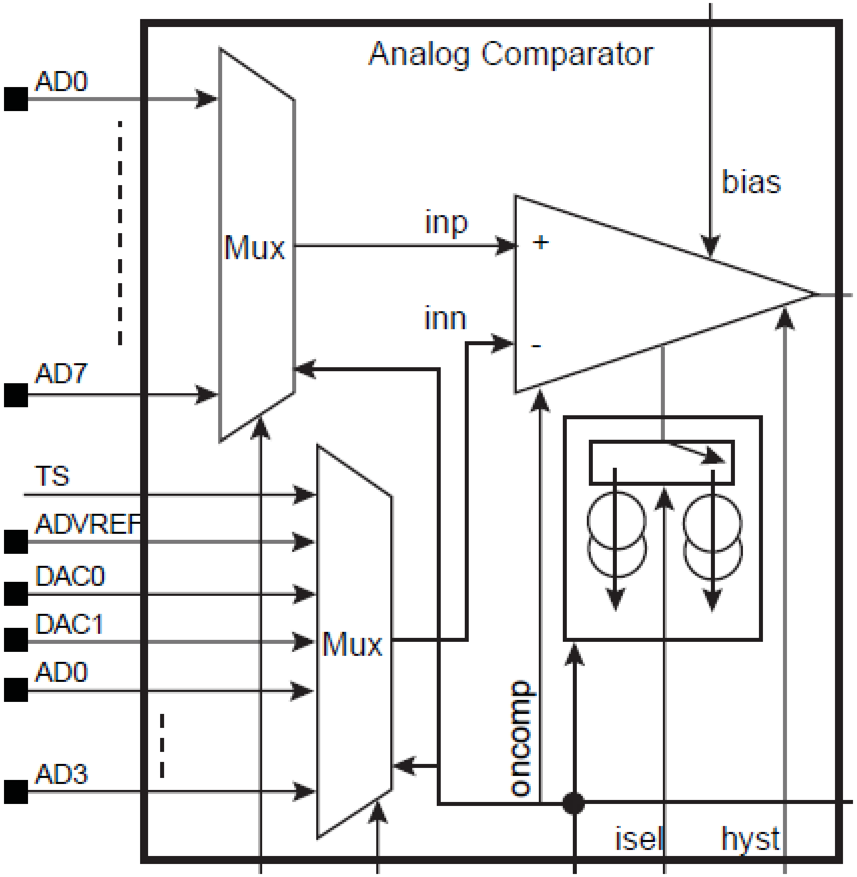
Pulse Width Modulation Controller (PWM)

- Synchronous Channel mode
  - Synchronous Channels share the same counter
  - Mode to update the synchronous channels registers after a programmable number of periods
- Two independent event lines which can send up to 4 triggers on ADC within a period
- One programmable Fault Input providing an asynchronous protection of outputs
- Stepper motor control (2 Channels)
- PDC Support
  - 1 Transmit PDC channel
  - Update duty cycle of synchronous channels

User Peripherals Highlights

Analog Comparator Controller (ACC)

- 8 User Analog Inputs Selectable for Comparison
  - AD0 to AD7
- 4 Voltage References Selectable for Comparison:
  - Temperature Sensor
  - ADVREF
  - DAC0 and DAC1
- Compare Event Fault Generation for PWM



User Peripherals Highlights

Analog-to-Digital Converter (ADC)

- Up to 16 Channels, 12-bit ADC
  - 10/12-bit resolution
  - Up to 1 MSample/s
- Programmable conversion sequence conversion on each channel
- Integrated temperature sensor (Analog Channel 15)
- Automatic calibration mode
- Single ended/differential conversion
- Programmable gain: 1, 2, 4
- PDC Support
  - 1 Receive PDC channel

User Peripherals Highlights

Digital-to-Analog Converter Controller (DACC)

- Up to Two Independent Analog Outputs, 12-bit Resolution
- Individual Enable and Disable of Each Analog Channel
- Hardware Trigger or External Trigger Pins
- Sleep Mode
- Automatic Wake-up on Trigger and Back-to-Sleep Mode after Conversions of all Enabled Channels
- PDC Support
    - 1 Transmit PDC channel

User Peripherals Highlights

Cyclic Redundancy Check Calculation Unit (CRCCU)

- 32-bit cyclic redundancy check automatic calculation
- Three different polynomials are available:
  - CCITT802.3
  - CASTAGNOLI
  - CCITT16.
- CRC comparison between two addresses of the memory
- One Master on the AHB with its own DMA Controller
  - Computes CRC on the fly

Atmel | Enabling Unlimited Possibilities®

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.