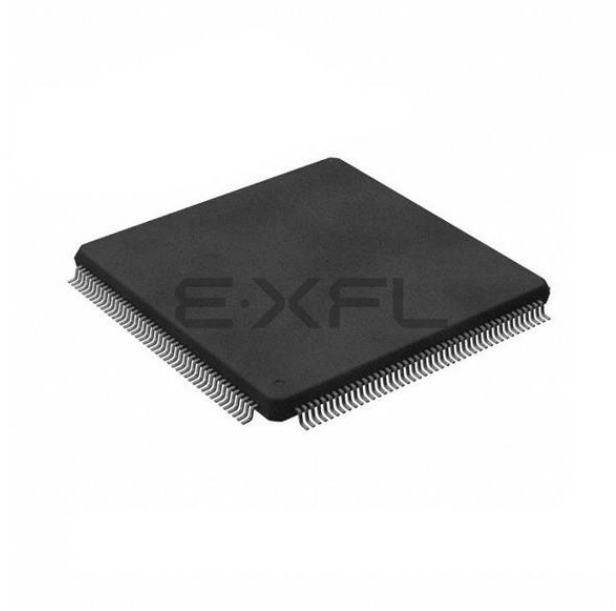


**Welcome to E-XFL.COM**

## What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

## Applications of "Embedded - Microcontrollers"

| Details | |
|---|---|
| Product Status | Obsolete |
| Core Processor | ARM7® |
| Core Size | 16/32-Bit |
| Speed | 33MHz |
| Connectivity | EBI/EMI, SPI, UART/USART |
| Peripherals | POR, WDT |
| Number of I/O | 58 |
| Program Memory Size | - |
| Program Memory Type | ROMless |
| EEPROM Size | - |
| RAM Size | 8K x 8 |
| Voltage - Supply (Vcc/Vdd) | 2.7V ~ 3.6V |
| Data Converters | A/D 8x10b; D/A 2x10b |
| Oscillator Type | Internal |
| Operating Temperature | -40°C ~ 85°C (TA) |
| Mounting Type | Surface Mount |
| Package / Case | 176-LQFP |
| Supplier Device Package | 176-TQFP (24x24) |
| Purchase URL | https://www.e-xfl.com/product-detail/microchip-technology/at91m55800a-33ai |

### 12.9.11 APMC Interrupt Enable Register

**Register Name:** APMC_IER

**Access Type:** Write-only

**Offset:** 0x34

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| – | – | – | – | – | – | – | – |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| – | – | – | – | – | – | – | – |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| – | – | – | – | – | – | LOCK | MOSCS |

- **MOSCS: Main Oscillator Interrupt Enable (Code Label `APMC_MOSCS`)**

0 = No effect.

1 = Enables the Main Oscillator Stabilized Interrupt.

- **LOCK: PLL Lock Interrupt Enable (Code Label `APMC_PLL_LOCK`)**

0 = No effect.

1 = Enables the PLL Lock Interrupt.

### 12.9.12 APMC Interrupt Disable Register

**Register Name:** APMC_IDR

**Access Type:** Write-only

**Offset:** 0x38

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| – | – | – | – | – | – | – | – |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| – | – | – | – | – | – | – | – |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| – | – | – | – | – | – | LOCK | MOSCS |

- **MOSCS: Main Oscillator Interrupt Disable (Code Label `APMC_MOSCS`)**

0 = No effect.

1 = Disables the Main Oscillator Stabilized Interrupt.

- **LOCK: PLL Lock Interrupt Disable (Code Label `APMC_PLL_LOCK`)**

0 = No effect.

1 = Disables the PLL Lock Interrupt.

### 13.3.12 RTC Valid Entry Register

**Register Name:** RTC_VER

**Access Type:** Read-only

**Reset State:** 0x0

**Offset:** 0x2C

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| – | – | – | – | – | – | – | – |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| – | – | – | – | – | – | – | – |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| – | – | – | – | NVCAL | NVTAL | NVC | NVT |

- **NVT: Non-Valid Time (Code Label RTC_NVT)**

0 = No invalid data has been detected in RTC_TIMR.

1 = RTC_TIMR has contained invalid data since it was last programmed.

- **NVC: Non-Valid Calendar (Code Label RTC_NVC)**

0 = No invalid data has been detected in RTC_CALR.

1 = RTC_CALR has contained invalid data since it was last programmed.

- **NVTAL: Non-Valid Time Alarm (Code Label RTC_NVTAL)**

0 = No invalid data has been detected in RTC_TAR.

1 = RTC_TAR has contained invalid data since it was last programmed.

- **NVCAL: Non-Valid Calendar Alarm (Code Label RTC_NVCAL)**

0 = No invalid data has been detected in RTC_CAR.

1 = RTC_CAR has contained invalid data since it was last programmed.

### 14.1.3 WD Control Register

**Name:** WD_CR

**Access:** Write-only

**Offset:** 0x08

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| – | – | – | – | – | – | – | – |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| RSTKEY | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| RSTKEY | | | | | | | |

- **RSTKEY: Restart Key (Code Label `WD_RSTKEY`)**

0xC071 = Watch Dog counter is restarted.

Other value = No effect.

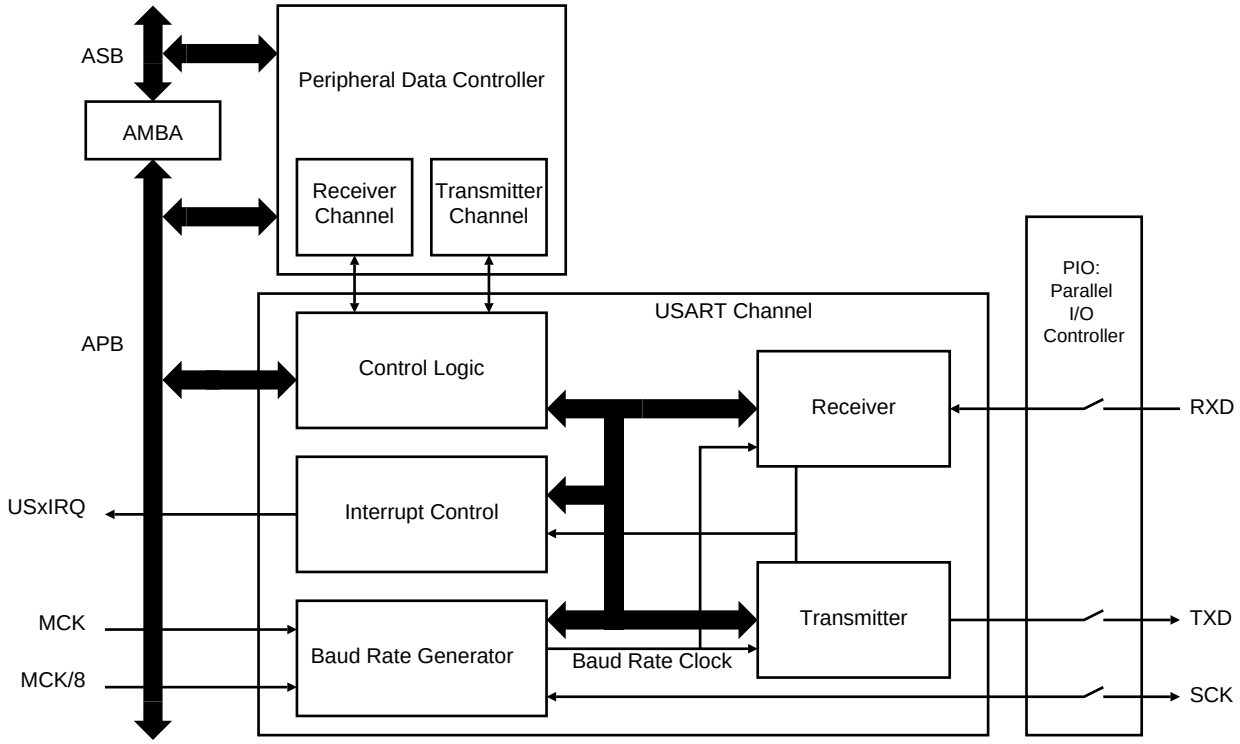
# 18. USART: Universal Synchronous/ Asynchronous Receiver/Transmitter

The AT91M55800AA provides three identical, full-duplex, universal synchronous/asynchronous receiver/transmitters which are connected to the Peripheral Data Controller.

The main features are:

- Programmable Baud Rate Generator
- Parity, Framing and Overrun Error Detection
- Line Break Generation and Detection
- Automatic Echo, Local Loopback and Remote Loopback channel modes
- Multi-drop Mode: Address Detection and Generation
- Interrupt Generation
- Two Dedicated Peripheral Data Controller channels
- 5-, 6-, 7-, 8- and 9-bit character length

**Figure 18-1.** USART Block Diagram



- Software Trigger: Each channel has a software trigger, available by setting SWTRG in TC_CCR.
- SYNC: Each channel has a synchronization signal SYNC. When asserted, this signal has the same effect as a software trigger. The SYNC signals of all channels are asserted simultaneously by writing TC_BCR (Block Control) with SYNC set.
- Compare RC Trigger: RC is implemented in each channel and can provide a trigger when the counter value matches the RC value if CPCTRG is set in TC_CMR.

The Timer Counter channel can also be configured to have an external trigger. In Capture Mode, the external trigger signal can be selected between TIOA and TIOB. In Waveform Mode, an external event can be programmed on one of the following signals: TIOB, XC0, XC1 or XC2. This external event can then be programmed to perform a trigger by setting ENETRG in TC_CMR.

If an external trigger is used, the duration of the pulses must be longer than the system clock (MCK) period in order to be detected.

### 22.2.2 DAC Mode Register

Register Name: DAC_MR
Access Type: Read/Write
Reset State: 0
Offset: 0x04

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| | | | | | | | |
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| | | | | | | | |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| | | | RES | TTRGSEL | | | TTRGEN |

TTRGEN: Timer Trigger Enable (Code Label **DAC_TTRGEN_EN**)

| TTRGEN | Selected TTRGEN | Code Label |
|---|---|---|
| 0 | The data written into the Data Holding Register (DAC_DHR) is transferred one main clock cycle later to the data output register (DAC_DOR). | DAC_TTRGEN_DIS |
| 1 | The data transfer from the DAC_DHR to the DAC_DOR is synchronized by the timer trigger. | DAC_TTRGEN_EN |

TTRGSEL: Timer Trigger Selection

Only used if TTRGEN = 1

| TTRGSEL | | | Selected Timer Trigger | Code Label |
|---|---|---|---|---|
| | | | | **DAC_TTRGSEL** |
| 0 | 0 | 0 | TIOA0 | DAC_TRG_TIOA0 |
| 0 | 0 | 1 | TIOA1 | DAC_TRG_TIOA1 |
| 0 | 1 | 0 | TIOA2 | DAC_TRG_TIOA2 |
| 0 | 1 | 1 | TIOA3 | DAC_TRG_TIOA3 |
| 1 | 0 | 0 | TIOA4 | DAC_TRG_TIOA4 |
| 1 | 0 | 1 | TIOA5 | DAC_TRG_TIOA5 |
| 1 | 1 | X | Reserved | |

RES: Resolution

| RES | Selected RES | Code Label |
|---|---|---|
| 0 | 10-bit resolution | DAC_10_BIT_RES |
| 1 | 8-bit resolution | DAC_8_BIT_RES |