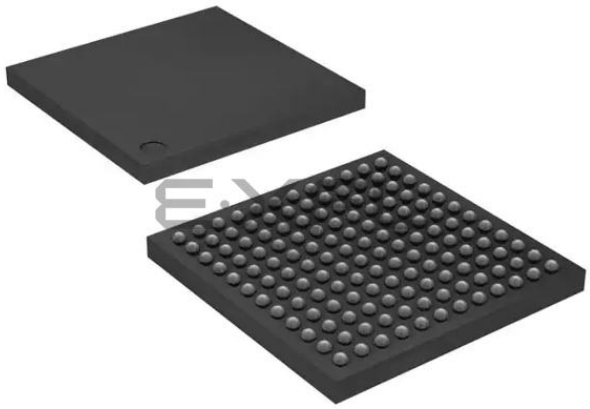


Welcome to E-XFL.COM

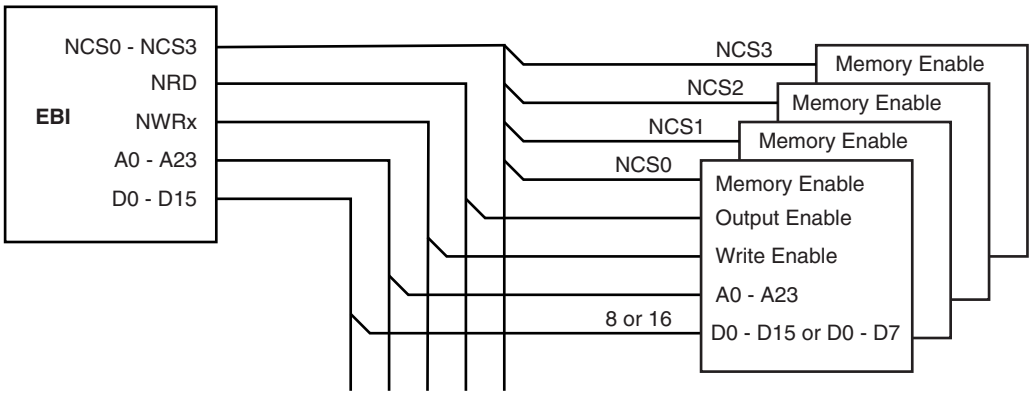
What is "Embedded - Microcontrollers"?

"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"

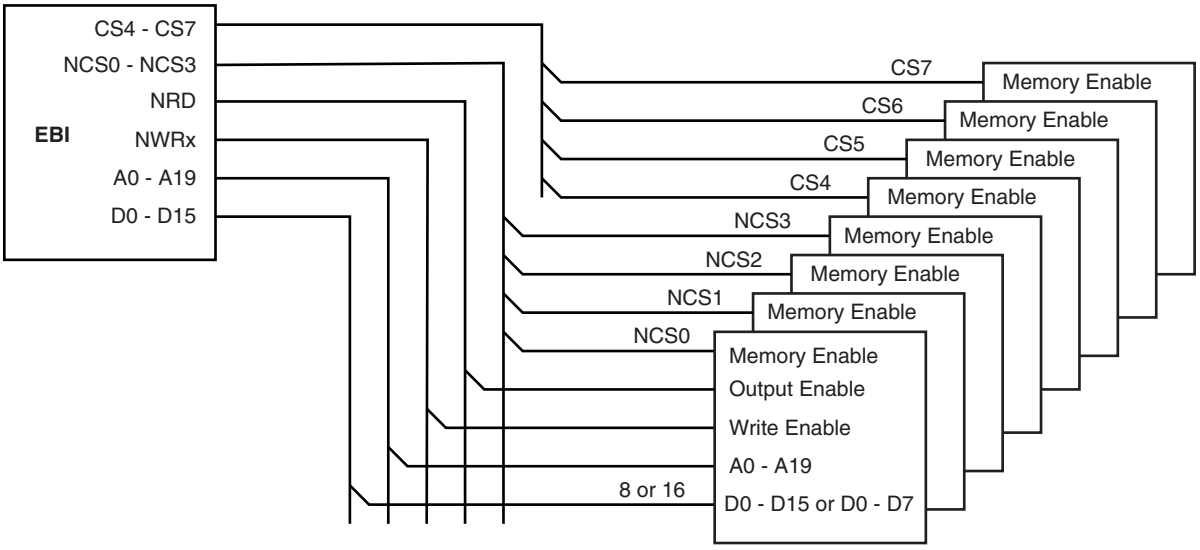
Details	
Product Status	Obsolete
Core Processor	ARM7®
Core Size	16/32-Bit
Speed	33MHz
Connectivity	EBI/EMI, SPI, UART/USART
Peripherals	WDT
Number of I/O	54
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	8K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	144-LFBGA
Supplier Device Package	144-BGA (13x13)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at91m42800a-33ci

Figure 11-2. Memory Connections for Four External Devices[1]



Notes: 1. For four external devices, the maximum address space per device is 16M bytes.

Figure 11-3. Memory Connections for Eight External Devices[1]



Notes: 1. For eight external devices, the maximum address space per device is 1M byte.

11.6 Data Bus Width

A data bus width of 8 or 16 bits can be selected for each chip select. This option is controlled by the DBW field in the EBI_CSR (Chip Select Register) for the corresponding chip select.

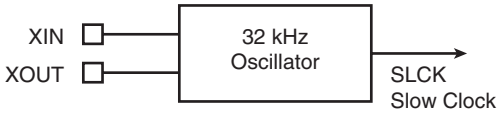
Figure 11-4 shows how to connect a 512K x 8-bit memory on NCS2.

Figures 11-21 through 11-27 show the timing cycles and wait states for read and write access to the various AT91M42800A external memory devices. The configurations described are shown in the following table:

Table 11-3. Memory Access Waveforms

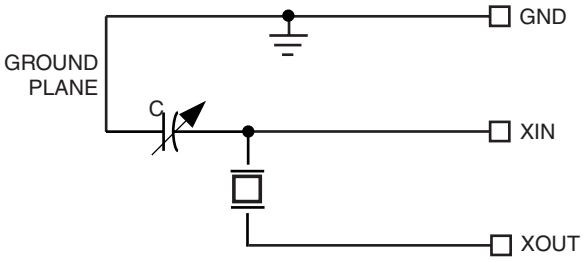
Figure Number	Number of Wait States	Bus Width	Size of Data Transfer
11-21	0	16	Word
11-22	1	16	Word
11-23	1	16	Half-word
11-24	0	8	Word
11-25	1	8	Half-word
11-26	1	8	Byte
11-27	0	16	Byte

Figure 12-2. Slow Clock



To operate correctly, the crystal must be as close to the XIN and XOUT pins as possible. An external variable capacitor can be added to adjust the oscillator frequency.

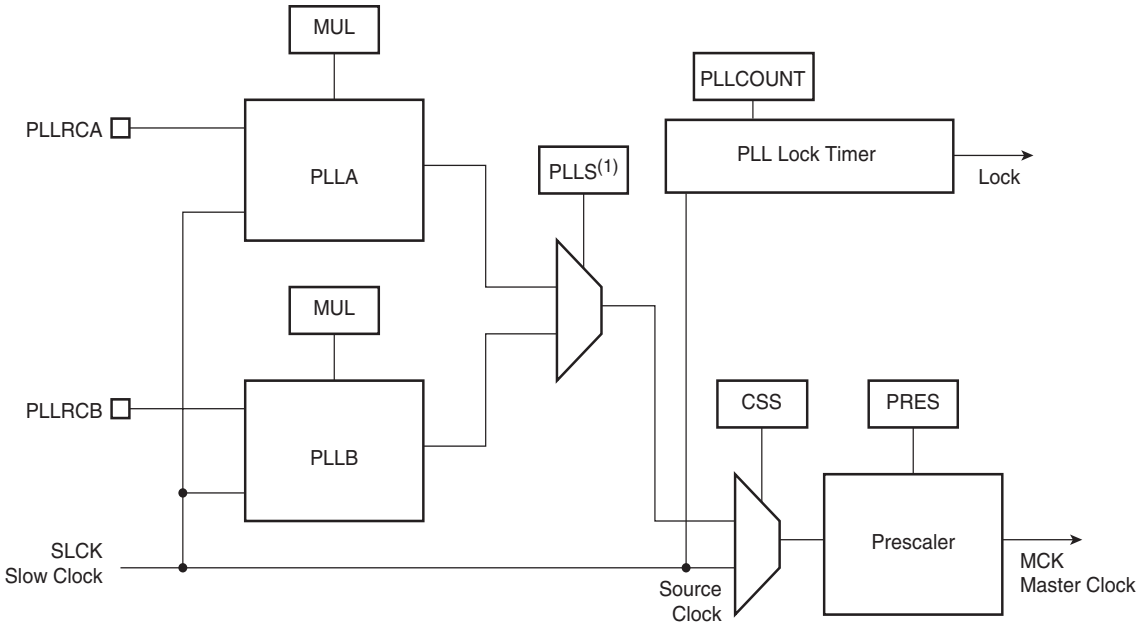
Figure 12-3. Crystal Location



12.2 Master Clock

The Master Clock (MCK) is generated from the Slow Clock by means of one of the two integrated PLLs and the prescaler.

Figure 12-4. Master Clock



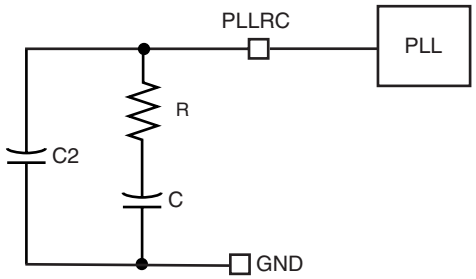
Note: 1. Value written at reset and not subsequently programmable.

12.2.1 Phase Locked Loops

Two PLLs are integrated in the AT91M42800A in order to cover a larger frequency range. Both PLLs have a Slow Clock input and a dedicated pin (PLLRCA or PLLRCB), which must have appropriate capacitors and resistors. The capacitors and resistors serve as a second

order filter. The PLLRC pin (A or B) that corresponds to the PLL that is disabled may be grounded if capacitors and resistors need to be saved.

Figure 12-5. PLL Capacitors and Resistors



Typical values for the two PLLs are shown below:

PLLA:

F_{SCLK} = 32.768 kHz

$F_{out_}$PLLA = 16.776 MHz

R = 1600 Ohm

C = 100 nF

C_2 = 10 nF

With these parameters, the output frequency is stable (±10%) in 600 µs. This settling time is the value to be programmed in the PLLCOUNT field of PMC_CGMR. The maximum frequency overshoot during this phase is 22.5 MHz.

PLLB:

F_{SCLK} = 32.768 kHz

$F_{out_}$PLLB = 33.554 MHz

R = 800 Ohm

C = 1 µF

C_2 = 100 nF

With these parameters, the output frequency is stable (±10%) in 4 ms. This settling time is the value to be programmed in the PLLCOUNT field of PMC_CGMR. The maximum frequency overshoot during this phase is 38 MHz.

12.2.2 PLL Selection

The required PLL must be selected at the first writing access and cannot be changed after that. The PLLS bit in PMC_CGMR (Clock Generator Mode Register) determines which PLL module is activated. The other PLL is disabled in order to reduce power consumption and can only be activated by another reset. Writing in PMC_CGMR with a different value has no effect.

12.2.3 Source Clock Selection

The bit CSS in PMC_CGMR selects the Slow Clock or the output of the activated PLL as the Source Clock of the prescaler. After reset, the CSS field is 0, selecting the Slow Clock as Source Clock.

When switching from Slow Clock to PLL Output, the Source Clock takes effect after 3 Slow Clock cycles plus 2.5 PLL output signal cycles. This is a maximum value.

12.14 PMC Status Register

Register Name: PMC_SR
Access Type: Read-only
Offset: 0x30
Reset Value: 0x0

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

- **LOCK: PLL Lock Status**

0 = The PLL output signal is not stabilized.

1 = The PLL output signal is stabilized.

loads the program counter with the interrupt handler address stored in the AIC_IVR register. Execution is then vectored to the interrupt handler corresponding to the current interrupt.

```
ldr PC,[PC,# -&F20]
```

The current interrupt is the interrupt with the highest priority when the Interrupt Vector Register (AIC_IVR) is read. The value read in the AIC_IVR corresponds to the address stored in the Source Vector Register (AIC_SVR) of the current interrupt. Each interrupt source has its corresponding AIC_SVR. In order to take advantage of the hardware interrupt vectoring it is necessary to store the address of each interrupt handler in the corresponding AIC_SVR, at system initialization.

14.2 Priority Controller

The NIRQ line is controlled by an 8-level priority encoder. Each source has a programmable priority level of 7 to 0. Level 7 is the highest priority and level 0 the lowest.

When the AIC receives more than one unmasked interrupt at a time, the interrupt with the highest priority is serviced first. If both interrupts have equal priority, the interrupt with the lowest interrupt source number (see Table 14-1) is serviced first.

The current priority level is defined as the priority level of the current interrupt at the time the register AIC_IVR is read (the interrupt which will be serviced).

In the case when a higher priority unmasked interrupt occurs while an interrupt already exists, there are two possible outcomes depending on whether the AIC_IVR has been read.

- If the NIRQ line has been asserted but the AIC_IVR has not been read, then the processor will read the new higher priority interrupt handler address in the AIC_IVR register and the current interrupt level is updated.
- If the processor has already read the AIC_IVR then the NIRQ line is reasserted. When the processor has authorized nested interrupts to occur and reads the AIC_IVR again, it reads the new, higher priority interrupt handler address. At the same time the current priority value is pushed onto a first-in last-out stack and the current priority is updated to the higher priority.

When the end of interrupt command register (AIC_EOICR) is written, the current interrupt level is updated with the last stored interrupt level from the stack (if any). Hence at the end of a higher priority interrupt, the AIC returns to the previous state corresponding to the preceding lower priority interrupt which had been interrupted.

14.3 Interrupt Handling

The interrupt handler must read the AIC_IVR as soon as possible. This de-asserts the NIRQ request to the processor and clears the interrupt in case it is programmed to be edge triggered. This permits the AIC to assert the NIRQ line again when a higher priority unmasked interrupt occurs.

At the end of the interrupt service routine, the end of interrupt command register (AIC_EOICR) must be written. This allows pending interrupts to be serviced.

14.4 Interrupt Masking

Each interrupt source, including FIQ, can be enabled or disabled using the command registers AIC_IECR and AIC_IDCR. The interrupt mask can be read in the Read-only register AIC_IMR. A disabled interrupt does not affect the servicing of other interrupts.

14.11 AIC Source Mode Register

Register Name: AIC_SMR0..AIC_SMR31
Access Type: Read/Write
Reset Value: 0

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
–	SRCTYPE		–	–	PRIOR		

- **PRIOR: Priority Level (Code Label `AIC_PRIOR`)**

Program the priority level for all sources except source 0 (FIQ).
The priority level can be between 0 (lowest) and 7 (highest).
The priority level is not used for the FIQ, in the SMR0.

- **SRCTYPE: Interrupt Source Type (Code Label `AIC_SRCTYPE`)**

Program the input to be positive or negative edge-triggered or positive or negative level sensitive.
The active level or edge is not programmable for the internal sources.

SRCTYPE		External Sources	Code Label
0	0	Low-level Sensitive	AIC_SRCTYPE_EXT_LOW_LEVEL
0	1	Negative Edge triggered	AIC_SRCTYPE_EXT_NEGATIVE_EDGE
1	0	High-level Sensitive	AIC_SRCTYPE_EXT_HIGH_LEVEL
1	1	Positive Edge triggered	AIC_SRCTYPE_EXT_POSITIVE_EDGE

SRCTYPE		Internal Sources	Code Label
X	0	Level Sensitive	AIC_SRCTYPE_INT_LEVEL_SENSITIVE
X	1	Edge triggered	AIC_SRCTYPE_INT_EDGE_TRIGGERED

15.9 PIO Enable Register

Register Name: PIO_PER
Access Type: Write-only
Offset: 0x00

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register is used to enable individual pins to be controlled by the PIO Controller instead of the associated peripheral. When the PIO is enabled, the associated peripheral (if any) is held at logic zero.

0 = No effect.

1 = Enables the PIO to control the corresponding pin (disables peripheral control of the pin).

15.10 PIO Disable Register

Register Name: PIO_PDR
Access Type: Write-only
Offset: 0x04

31	30	29	28	27	26	25	24
P31	P30	P29	P28	P27	P26	P25	P24
23	22	21	20	19	18	17	16
P23	P22	P21	P20	P19	P18	P17	P16
15	14	13	12	11	10	9	8
P15	P14	P13	P12	P11	P10	P9	P8
7	6	5	4	3	2	1	0
P7	P6	P5	P4	P3	P2	P1	P0

This register is used to disable PIO control of individual pins. When the PIO control is disabled, the normal peripheral function is enabled on the corresponding pin.

0 = No effect.

1 = Disables PIO control (enables peripheral control) on the corresponding pin.

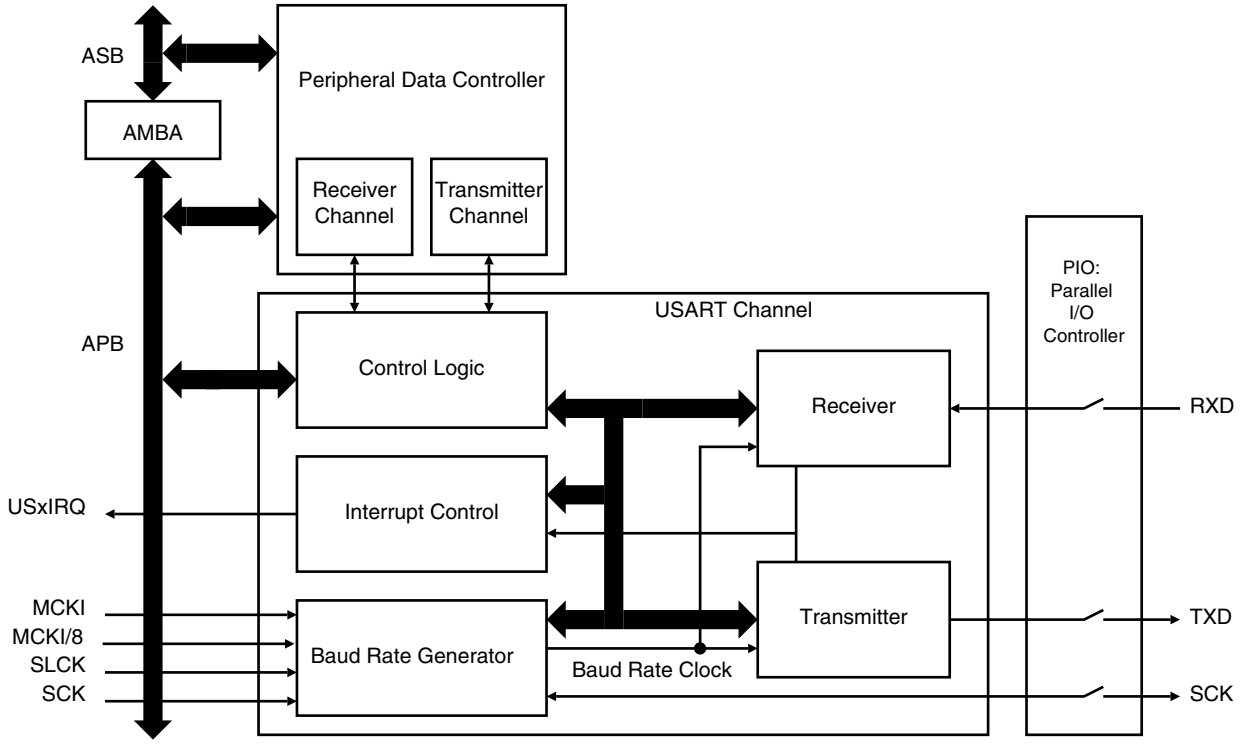
17. USART: Universal Synchronous/Asynchronous Receiver/Transmitter

The AT91M42800A provides two identical, full-duplex, universal synchronous/asynchronous receiver/transmitters that interface to the APB and are connected to the Peripheral Data Controller.

The main features are:

- Programmable Baud Rate Generator with External or Internal Clock, as well as Slow Clock
- 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 17-1. USART Block Diagram



The interpretation of the number of stop bits depends on SYNC.

NBSTOP		Asynchronous (SYNC = 0)	Synchronous (SYNC = 1)	Code Label: US_NBSTOP
0	0	1 stop bit	1 stop bit	US_NBSTOP_1
0	1	1.5 stop bits	Reserved	US_NBSTOP_1_5
1	0	2 stop bits	2 stop bits	US_NBSTOP_2
1	1	Reserved	Reserved	–

Note: 1.5 or 2 stop bits are reserved for the TX function. The RX function uses only the 1 stop bit (there is no check on the 2 stop bit timeslot if NBSTO P= 10).

- **CHMODE: Channel Mode**

CHMODE		Mode Description	Code Label: US_CHMODE
0	0	Normal Mode The USART Channel operates as an Rx/Tx USART.	US_CHMODE_NORMAL
0	1	Automatic Echo Receiver Data Input is connected to TXD pin.	US_CHMODE_AUTOMATIC_ECHO
1	0	Local Loopback Transmitter Output Signal is connected to Receiver Input Signal.	US_CHMODE_LOCAL_LOOPBACK
1	1	Remote Loopback RXD pin is internally connected to TXD pin.	US_CHMODE_REMOTE_LOOPBACK

- **MODE9: 9-Bit Character Length (Code Label US_MODE9)**

0 = CHRL defines character length.

1 = 9-Bit character length.

- **CKLO: Clock Output Select (Code Label US_CLKO)**

0 = The USART does not drive the SCK pin.

1 = The USART drives the SCK pin.

- **COMMTX: Disable ARM7TDMI ICE Debug Communication Channel Transmit Interrupt**

This bit is implemented for USART0 only.

0 = No effect.

1 = Disables COMMTX Interrupt.

- **COMMRX: Disable ARM7TDMI ICE Debug Communication Channel Receive Interrupt**

This bit is implemented for USART0 only.

0 = No effect.

1 = Disables COMMRX Interrupt.

17.17 USART Receiver Holding Register

Name: US_RHR

Access Type: Read-only

Offset: 0x18

Reset Value: 0x0

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

- **RXCHR: Received Character**

Last character received if RXRDY is set. When number of data bits is less than 9 bits, the bits are right-aligned.

All unused bits read zero.

17.18 USART Transmitter Holding Register

Name: US_THR

Access Type: Write-only

Offset: 0x1C

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

- **TXCHR: Character to be Transmitted**

Next character to be transmitted after the current character if TXRDY is not set. When number of data bits is less than 9 bits, the bits are right-aligned.

17.20 USART Receiver Time-out Register

Name: US_RTOR

Access Type: Read/Write

Offset: 0x24

Reset Value: 0x0

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
TO							

- **TO: Time-out Value**

When a value is written to this register, a Start Time-out Command is automatically performed.

TO	
0	Disables the RX Time-out function.
1- 255	The Time-out counter is loaded with TO when the Start Time-out Command is given or when each new data character is received (after reception has started).

Time-out duration = TO x 4 x Bit period

18. TC: Timer/Counter

The AT91M42800A features two Timer/Counter blocks, each containing three identical 16-bit Timer/Counter channels. Each channel can be independently programmed to perform a wide range of functions including frequency measurement, event counting, interval measurement, pulse generation, delay timing and pulse width modulation.

Each Timer/Counter (TC) channel has 3 external clock inputs, 5 internal clock inputs, and 2 multi-purpose input/output signals which can be configured by the user. Each channel drives an internal interrupt signal which can be programmed to generate processor interrupts via the AIC (Advanced Interrupt Controller).

The Timer/Counter block has two global registers which act upon all three TC channels. The Block Control Register allows the three channels to be started simultaneously with the same instruction. The Block Mode Register defines the external clock inputs for each Timer/Counter channel, allowing them to be chained.

Each Timer/Counter block operates independently and has a complete set of block and channel registers. Since they are identical in operation, only one block is described below (see Timer/Counter Description on page 152). The internal configuration of a single Timer/Counter Block is shown in Figure 18-1.

18.8 TC Channel Control Register

Register Name: TC_CCR

Access Type: Write-only

Offset: 0x00

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
–	–	–	–	–	SWTRG	CLKDIS	CLKEN

- **CLKEN: Counter Clock Enable Command (Code Label `TC_CLKEN`)**

0 = No effect.

1 = Enables the clock if CLKDIS is not 1.

- **CLKDIS: Counter Clock Disable Command (Code Label `TC_CLKDIS`)**

0 = No effect.

1 = Disables the clock.

- **SWTRG: Software Trigger Command (Code Label `TC_SWTRG`)**

0 = No effect.

1 = A software trigger is performed: the counter is reset and clock is started.

18.9 TC Channel Mode Register: Capture Mode

Register Name: TC_CMR

Access Type: Read/Write

Offset: 0x04

Reset Value: 0x0

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	–	–	–	LDRB		LDRA	
15	14	13	12	11	10	9	8
WAVE=0	CPCTRG	–	–	–	ABETRG	ETRGEDG	
7	6	5	4	3	2	1	0
LDBDIS	LDBSTOP	BURST		CLKI	TCCLKS		

- **TCCLKS: Clock Selection**

TCCLKS			Clock Selected	Code Label: `TC_CLKS`
0	0	0	MCK/2	`TC_CLKS_MCK2`
0	0	1	MCK/8	`TC_CLKS_MCK8`
0	1	0	MCK/32	`TC_CLKS_MCK32`
0	1	1	MCK/128	`TC_CLKS_MCK128`
1	0	0	SLCK	`TC_CLKS_SLCK`
1	0	1	XC0	`TC_CLKS_XC0`
1	1	0	XC1	`TC_CLKS_XC1`
1	1	1	XC2	`TC_CLKS_XC2`

- **CLKI: Clock Invert (Code Label `TC_CLKI`)**

0 = Counter is incremented on rising edge of the clock.

1 = Counter is incremented on falling edge of the clock.

- **BURST: Burst Signal Selection**

BURST		Selected BURST	Code Label: `TC_BURST`
0	0	The clock is not gated by an external signal.	`TC_BURST_NONE`
0	1	XC0 is ANDed with the selected clock.	`TC_BURST_XC0`
1	0	XC1 is ANDed with the selected clock.	`TC_BURST_XC1`
1	1	XC2 is ANDed with the selected clock.	`TC_BURST_XC2`

- **LDBSTOP: Counter Clock Stopped with RB Loading (Code Label `TC_LDBSTOP`)**

0 = Counter clock is not stopped when RB loading occurs.

1 = Counter clock is stopped when RB loading occurs.

- **LDBDIS: Counter Clock Disable with RB Loading (Code Label `TC_LDBDIS`)**

0 = Counter clock is not disabled when RB loading occurs.

1 = Counter clock is disabled when RB loading occurs.

- **AEEVT: External Event Effect on TIOA**

AEEVT		Effect	Code Label: TC_AEEVT
0	0	None	TC_AEEVT_OUTPUT_NONE
0	1	Set	TC_AEEVT_SET_OUTPUT
1	0	Clear	TC_AEEVT_CLEAR_OUTPUT
1	1	Toggle	TC_AEEVT_TOGGLE_OUTPUT

- **ASWTRG: Software Trigger Effect on TIOA**

ASWTRG		Effect	Code Label: TC_ASWTRG
0	0	None	TC_ASWTRG_OUTPUT_NONE
0	1	Set	TC_ASWTRG_SET_OUTPUT
1	0	Clear	TC_ASWTRG_CLEAR_OUTPUT
1	1	Toggle	TC_ASWTRG_TOGGLE_OUTPUT

- **BCPB: RB Compare Effect on TIOB**

BCPB		Effect	Code Label: TC_BCPB
0	0	None	TC_BCPB_OUTPUT_NONE
0	1	Set	TC_BCPB_SET_OUTPUT
1	0	Clear	TC_BCPB_CLEAR_OUTPUT
1	1	Toggle	TC_BCPB_TOGGLE_OUTPUT

- **BCPC: RC Compare Effect on TIOB**

BCPC		Effect	Code Label: TC_BCPC
0	0	None	TC_BCPC_OUTPUT_NONE
0	1	Set	TC_BCPC_SET_OUTPUT
1	0	Clear	TC_BCPC_CLEAR_OUTPUT
1	1	Toggle	TC_BCPC_TOGGLE_OUTPUT

- **BEEVT: External Event Effect on TIOB**

BEEVT		Effect	Code Label: TC_BEEVT
0	0	None	TC_BEEVT_OUTPUT_NONE
0	1	Set	TC_BEEVT_SET_OUTPUT
1	0	Clear	TC_BEEVT_CLEAR_OUTPUT
1	1	Toggle	TC_BEEVT_TOGGLE_OUTPUT

19.18 SPI Transmit Pointer Register

Name: SP_TPR

Access Type: Read/Write

Offset: 0x28

Reset Value: 0x0

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

- **TXPTR: Transmit Pointer**

TXPTR must be loaded with the address of the transmit buffer.

19.19 SPI Transmit Counter Register

Name: SP_TCR

Access Type: Read/Write

Offset: 0x2C

Reset Value: 0x0

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

- **TXCTR: Transmit Counter**

TXCTR must be loaded with the size of the transmit buffer.

0: Stop Peripheral Data Transfer dedicated to the transmitter.

1-65535: Start Peripheral Data transfer if TDRE is active.

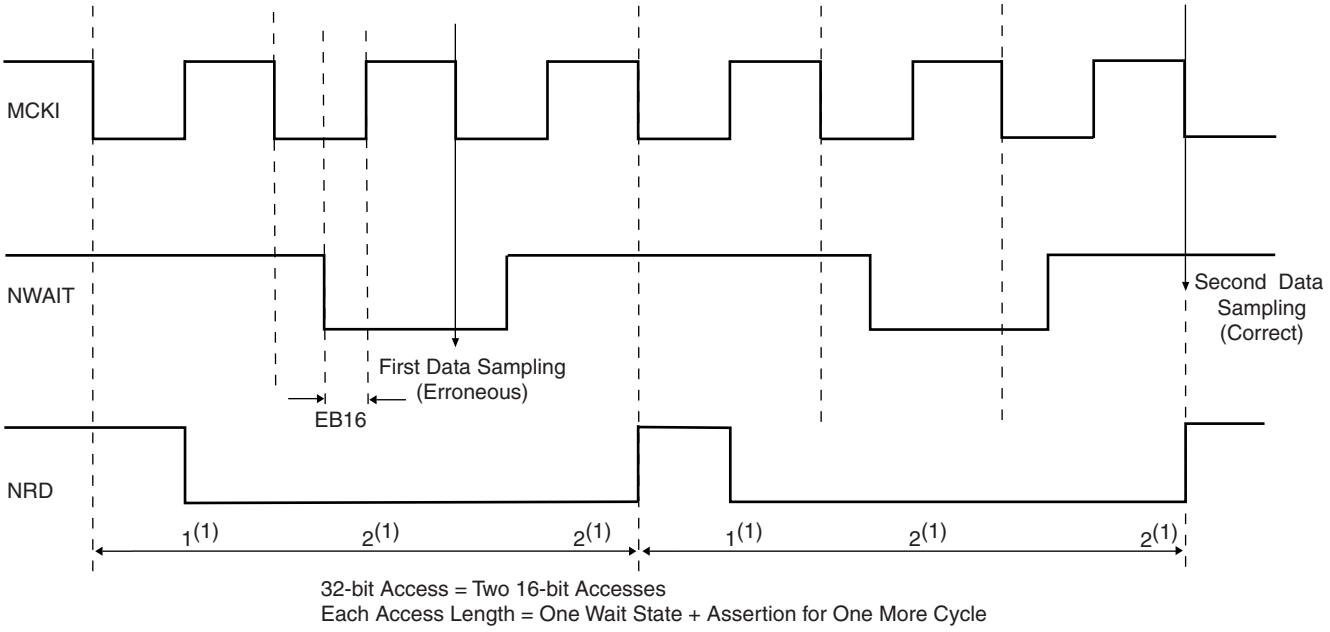
23. Ordering Information

Table 23-1. Ordering Information

Ordering Code	Package	Package Type	Operating Temperature Range
AT91M42800A-33CJ	BGA 144	RoHS-compliant	Industrial (-40° C to 85° C)
AT91M42800A-33AU	LQFP 144	Green	

If the first two conditions are not met during a 32-bit read access, the first 16-bit data is read at the end of the standard 16-bit read access. In the following example, the number of standard waits is one. NWAIT assertions do affect both NRD pulse lengths, but first data sampling is not delayed. The second data sampling is correct.

Figure 24-3. Number of Standard Wait States is One



Note: 1. These numbers refer to the standard access cycles.

If the first two conditions are not met during write accesses, the NWE signal is not affected by the NWAIT assertion. The following example illustrates the number of standard wait states. NWAIT is not asserted during the first cycle, but is asserted at the second and last cycle of the standard access. The access is correctly delayed as the NCS line rises accordingly to the NWAIT assertion. However, the NWE signal waveform is unchanged, and rises too early.