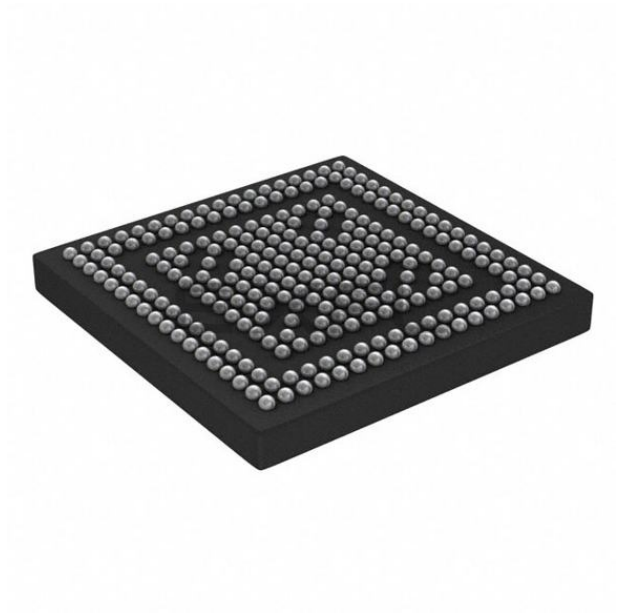


Welcome to E-XFL.COM

Understanding Embedded - Microprocessors

Embedded microprocessors are specialized computing chips designed to perform specific tasks within an embedded system. Unlike general-purpose microprocessors found in personal computers, embedded microprocessors are tailored for dedicated functions within larger systems, offering optimized performance, efficiency, and reliability. These microprocessors are integral to the operation of countless electronic devices, providing the computational power necessary for controlling processes, handling data, and managing communications.

Applications of Embedded - Microprocessors

Embedded microprocessors are utilized across a broad spectrum of applications, making them indispensable in

Details	
Product Status	Obsolete
Core Processor	ARM® Cortex®-A5
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	500MHz
Co-Processors/DSP	Multimedia; NEON™ MPE
RAM Controllers	LPDDR1, LPDDR2, LPDDR3, DDR2, DDR3, DDR3L, QSPI
Graphics Acceleration	Yes
Display & Interface Controllers	Keyboard, LCD, Touchscreen
Ethernet	10/100Mbps (1)
SATA	-
USB	USB 2.0 + HSIC
Voltage - I/O	3.3V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	ARM TZ, Boot Security, Cryptography, RTIC, Secure Fusebox, Secure JTAG, Secure Memory, Secure RTC
Package / Case	256-TFBGA
Supplier Device Package	256-TFBGA (8x8)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsama5d24a-cu

14.5.28 L2CC Prefetch Control Register

Name: L2CC_PCR

Address: 0x00A00F60

Access: Read/Write in Secure mode

Read-only in Non-secure mode

31	30	29	28	27	26	25	24
–	DLEN	INSPEN	DATPEN	DLFWRDIS	–	–	PDEN
23	22	21	20	19	18	17	16
IDLEN	–	NSIDEN	–	–	–	–	–
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
7	6	5	4	3	2	1	0
–	–	–	OFFSET				

OFFSET: Prefetch Offset

You must only use the Prefetch offset values of 0-7, 15, 23, and 31 for these bits. The L2CC does not support the other values.

NSIDEN: Not Same ID on Exclusive Sequence Enable

0: Read and write portions of a non-cacheable exclusive sequence have the same AXI ID when issued to L3. This is the default value.

1: Read and write portions of a non-cacheable exclusive sequence do not have the same AXI ID when issued to L3.

IDLEN: INCR Double Linefill Enable

0: The L2CC does not issue INCR 8x64-bit read bursts to L3 on reads that miss in the L2 cache. This is the default value.

1: The L2CC can issue INCR 8x64-bit read bursts to L3 on reads that miss in the L2 cache.

Note: This bit can only be used if the DLEN bit is set HIGH. See Section 14.4.1 “Double Linefill Issuing” for details on double linefill functionality.

PDEN: Prefetch Drop Enable

0: The L2CC does not discard prefetch reads issued to L3. This is the default value.

1: The L2CC discards prefetch reads issued to L3 when there is a resource conflict with explicit reads.

DLFWRDIS: Double Linefill on WRAP Read Disable

0: Double linefill on WRAP read is enabled. This is the default value.

1: Double linefill on WRAP read is disabled.

Note: This bit can only be used if the DLEN bit is set HIGH. See Section 14.4.1 “Double Linefill Issuing” for details on double linefill functionality.

DATPEN: Data Prefetch Enable

0: Data prefetching is disabled. This is the default value.

1: Data prefetching is enabled.

INSPEN: Instruction Prefetch Enable

0: Instruction prefetching is disabled. This is the default value.

1: Instruction prefetching is enabled.

DLEN: Double Linefill Enable

0: The L2CC always issues 4x64-bit read bursts to L3 on reads that miss in the L2 cache. This is the default value.

1: The L2CC issues 8x64-bit read bursts to L3 on reads that miss in the L2 cache.

21.9.5 AIC FIQ Vector Register

Name: AIC_FVR

Address: 0xFC020014 (AIC), 0xF803C014 (SAIC)

Access: Read-only

31	30	29	28	27	26	25	24
FIQV							

23	22	21	20	19	18	17	16
FIQV							

15	14	13	12	11	10	9	8
FIQV							

7	6	5	4	3	2	1	0
FIQV							

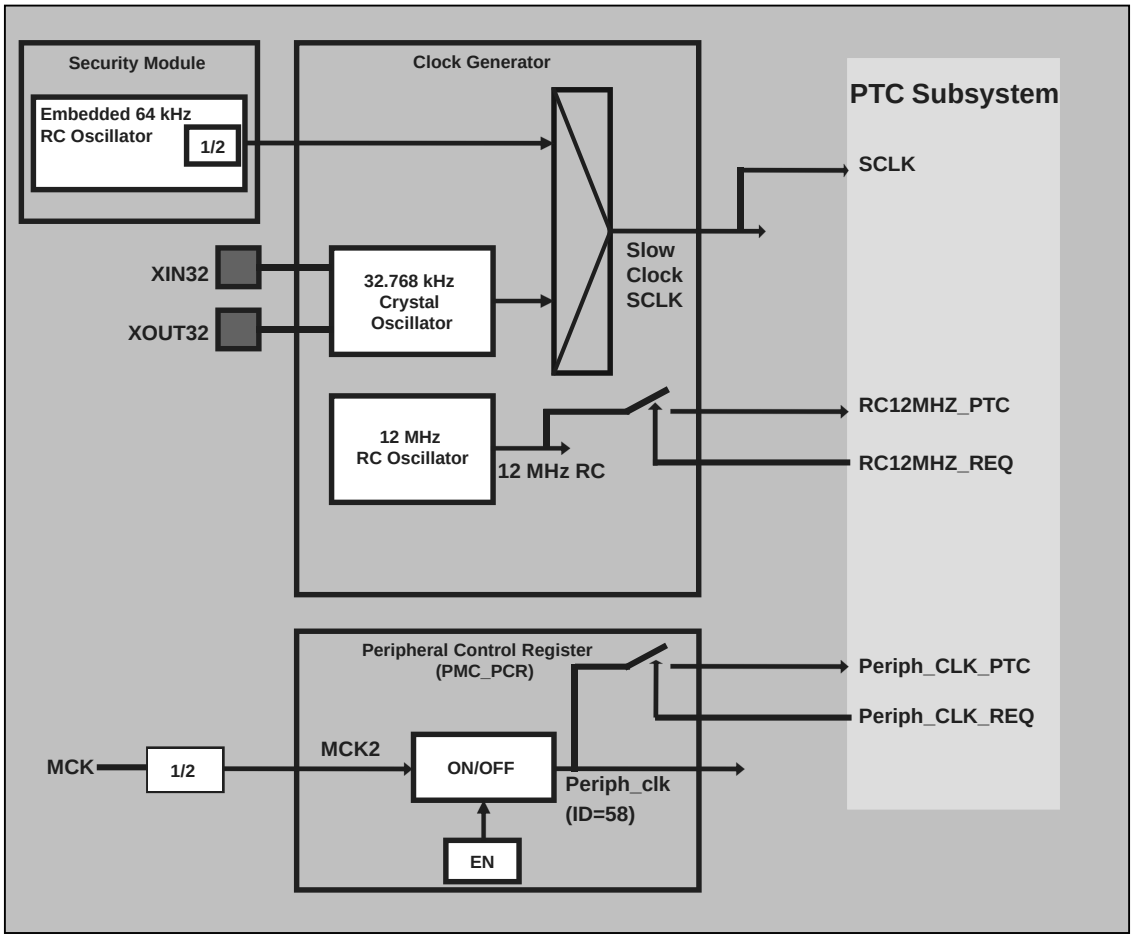
FIQV: FIQ Vector Register

The FIQ Vector Register contains the vector programmed by the user in the Source Vector Register when INTSEL = 0. When there is no fast interrupt, the FIQ Vector Register reads the value stored in AIC_SPU.

29.5.1 Power Management

The PTC Controller is not continuously clocked. The programmer must first enable the PTC Controller peripheral clock in the Power Management Controller (PMC) before using the PTC Controller. However, if the application does not require PTC operations, the PTC Controller clock can be stopped when not needed and restarted when necessary. Configuring the PTC Controller requires the PTC Controller clock to be enabled.

Figure 29-2: PTC Subsystem Clock Sources



The PTC subsystem operates both from a peripheral clock synchronous to the master clock of the system and from an asynchronous clock source directly connected to the embedded 12 MHz RC oscillator. The selected clocks must be enabled in the PMC before they can be used by the PTC. By default, the 12 MHz RC oscillator is enabled at startup of the product.

The various clock sources are as follows:

- PERIPH_CLK_PTC

This clock source is dedicated to the picoPower processor. It is located in the PMC as Periph_clk[PID]=PCLOCK_LS. This clock is synchronous with the AHB/APB matrix controlling the host interface and the mailbox. The clock frequency is between 12 MHz and 83 MHz. The same clock is used for the ARM interface connected as an APB slave via an AHB/APB bridge. It is also used to program the code/data SRAM and to access the mailbox SRAM.

- RC12MHZ

A different clock is used for the PTC digital controller. This clock can be divided internally in the pPP before being used. There is also a small local prescaler in the PTC digital controller to allow lower clock rates. Thus, the PTC operates from an asynchronous clock source and the operation is independent from the main system clock and its derivative clocks, such as the peripheral bus clock (PERIPH_CLK_PTC).

- SCLK

For the timers, a 32 kHz clock is used and divided internally down to a 1 kHz clock for counting the timer interrupt.

34.7.20 Secure PIO Clear Output Data Register

Name: S_PIO_CODRx [x=0..3]

Address: 0xFC039014 [0], 0xFC039054 [1], 0xFC039094 [2], 0xFC0390D4 [3]

Access: Write-only

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

P0–P31: Clear Output Data

0: No effect.

1: Clears the data to be driven on the I/O line of the I/O group x.

Table 39-55: Register Mapping (Continued)

Offset	Register	Name	Access	Reset
0x0000006C	Base Layer Configuration Register 0	LCDC_BASECFG0	Read/ Write	0x00000000
0x00000070	Base Layer Configuration Register 1	LCDC_BASECFG1	Read/ Write	0x00000000
0x00000074	Base Layer Configuration Register 2	LCDC_BASECFG2	Read/ Write	0x00000000
0x00000078	Base Layer Configuration Register 3	LCDC_BASECFG3	Read/ Write	0x00000000
0x0000007C	Base Layer Configuration Register 4	LCDC_BASECFG4	Read/ Write	0x00000000
0x00000080	Base Layer Configuration Register 5	LCDC_BASECFG5	Read/ Write	0x00000000
0x00000084	Base Layer Configuration Register 6	LCDC_BASECFG6	Read/ Write	0x00000000
0x00000088–0x0000013C	Reserved	–	–	–
0x00000140	Overlay 1 Channel Enable Register	LCDC_OVR1CHER	Write-only	–
0x00000144	Overlay 1 Channel Disable Register	LCDC_OVR1CHDR	Write-only	–
0x00000148	Overlay 1 Channel Status Register	LCDC_OVR1CHSR	Read-only	0x00000000
0x0000014C	Overlay 1 Interrupt Enable Register	LCDC_OVR1IER	Write-only	–
0x00000150	Overlay 1 Interrupt Disable Register	LCDC_OVR1IDR	Write-only	–
0x00000154	Overlay 1 Interrupt Mask Register	LCDC_OVR1IMR	Read-only	0x00000000
0x00000158	Overlay 1 Interrupt Status Register	LCDC_OVR1ISR	Read-only	0x00000000
0x0000015C	Overlay 1 DMA Head Register	LCDC_OVR1HEAD	Read/ Write	0x00000000
0x00000160	Overlay 1 DMA Address Register	LCDC_OVR1ADDR	Read/ Write	0x00000000
0x00000164	Overlay 1 DMA Control Register	LCDC_OVR1CTRL	Read/ Write	0x00000000
0x00000168	Overlay 1 DMA Next Register	LCDC_OVR1NEXT	Read/ Write	0x00000000
0x0000016C	Overlay 1 Configuration Register 0	LCDC_OVR1CFG0	Read/ Write	0x00000000
0x00000170	Overlay 1 Configuration Register 1	LCDC_OVR1CFG1	Read/ Write	0x00000000
0x00000174	Overlay 1 Configuration Register 2	LCDC_OVR1CFG2	Read/ Write	0x00000000
0x00000178	Overlay 1 Configuration Register 3	LCDC_OVR1CFG3	Read/ Write	0x00000000
0x0000017C	Overlay 1 Configuration Register 4	LCDC_OVR1CFG4	Read/ Write	0x00000000
0x00000180	Overlay 1 Configuration Register 5	LCDC_OVR1CFG5	Read/ Write	0x00000000

PPIE: Post Processing Interrupt Enable

0: No effect.

1: Enables the interrupt.

39.7.13 LCD Controller Interrupt Mask Register

Name: LCDC_LCDIMR

Address: 0xF0000034

Access: Read-only

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

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

15	14	13	12	11	10	9	8
–	–	PPIM	–	HEOIM	OVR2IM	OVR1IM	BASEIM

7	6	5	4	3	2	1	0
–	–	–	FIFOERRIM	–	DISPIM	DISIM	SOFIM

SOFIM: Start of Frame Interrupt Mask

0: Interrupt source is disabled.

1: Interrupt source is enabled.

DISIM: LCD Disable Interrupt Mask

0: Interrupt source is disabled.

1: Interrupt source is enabled.

DISPIM: Powerup/Powerdown Sequence Terminated Interrupt Mask

0: Interrupt source is disabled.

1: Interrupt source is enabled.

FIFOERRIM: Output FIFO Error Interrupt Mask

0: Interrupt source is disabled.

1: Interrupt source is enabled.

BASEIM: Base Layer Interrupt Mask

0: Interrupt source is disabled.

1: Interrupt source is enabled.

OVR1IM: Overlay 1 Interrupt Mask

0: Interrupt source is disabled.

1: Interrupt source is enabled.

OVR2IM: Overlay 2 Interrupt Mask

0: Interrupt source is disabled.

1: Interrupt source is enabled.

HEOIM: High-End Overlay Interrupt Mask

0: Interrupt source is disabled.

1: Interrupt source is enabled.

PPIM: Post Processing Interrupt Mask

0: Interrupt source is disabled

1: Interrupt source is enabled

40.8.41 GMAC 1588 Timer Second Comparison Low Register

Name: GMAC_SCL

Address: 0xF80080E0

Access: Read/Write

31	30	29	28	27	26	25	24
SEC							

23	22	21	20	19	18	17	16
SEC							

15	14	13	12	11	10	9	8
SEC							

7	6	5	4	3	2	1	0
SEC							

SEC: 1588 Timer Second Comparison Value

Value is compared to seconds value bits [31:0] of the TSU timer count value.

40.8.81 GMAC Oversized Frames Received Register

Name: GMAC_OFR

Address: 0xF8008188

Access: Read-only

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

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

15	14	13	12	11	10	9	8
–	–	–	–	–	–	OFRX	

7	6	5	4	3	2	1	0
OFRX							

OFRX: Oversized Frames Received

This register counts the number of frames received exceeding 1518 bytes (1536 bytes if bit 8 is set in the Network Configuration Register) in length but do not have either a CRC error, an alignment error nor a receive symbol error. See Section 40.8.2 ”GMAC Network Configuration Register”.

This field determines the current number of bytes still to be transferred for this buffer.

This field is decremented from the AHB source bus access byte width at the end of this bus address phase.

The access byte width is 4 by default, or less, at DMA start or end, if the start or end address is not aligned on a word boundary.

At the end of buffer, the DMA accesses the UDPHS device only for the number of bytes needed to complete it.

This field value is reliable (stable) only if the channel has been stopped or frozen (UDPHS_EPTCTLx register NT_DIS_DMA bit is used to disable the channel request) and the channel is no longer active CHANN_ACT flag is 0.

Note: For OUT endpoints, if the receive buffer byte length (BUFF_LENGTH) has been defaulted to zero because the USB transfer length is unknown, the actual buffer byte length received is 0x10000-BUFF_COUNT.

45.7 Product Dependencies

45.7.1 I/O Lines

The pins used for interfacing the compliant external devices may be multiplexed with PIO lines.

Before using the SSC receiver, the PIO controller must be configured to dedicate the SSC receiver I/O lines to the SSC Peripheral mode.

Before using the SSC transmitter, the PIO controller must be configured to dedicate the SSC transmitter I/O lines to the SSC Peripheral mode.

Table 45-2: I/O Lines

Instance	Signal	I/O Line	Peripheral
SSC0	RD0	PB23	C
SSC0	RD0	PC15	E
SSC0	RF0	PB25	C
SSC0	RF0	PC17	E
SSC0	RK0	PB24	C
SSC0	RK0	PC16	E
SSC0	TD0	PB22	C
SSC0	TD0	PC14	E
SSC0	TF0	PB21	C
SSC0	TF0	PC13	E
SSC0	TK0	PB20	C
SSC0	TK0	PC12	E
SSC1	RD1	PA17	B
SSC1	RD1	PB17	C
SSC1	RF1	PA19	B
SSC1	RF1	PB19	C
SSC1	RK1	PA18	B
SSC1	RK1	PB18	C
SSC1	TD1	PA16	B
SSC1	TD1	PB16	C
SSC1	TF1	PA15	B
SSC1	TF1	PB15	C
SSC1	TK1	PA14	B
SSC1	TK1	PB14	C

45.7.2 Power Management

The SSC is not continuously clocked. The SSC interface may be clocked through the Power Management Controller (PMC), therefore the programmer must first configure the PMC to enable the SSC clock.

45.7.3 Interrupt

The SSC interface has an interrupt line connected to the interrupt controller. Handling interrupts requires programming the interrupt controller before configuring the SSC.

When the start frame delimiter is a sync pattern (ONEBIT = 0), both command and data delimiter are supported. If a valid sync is detected, the received character is written as RXCHR field in the Receive Holding Register (FLEX_US_RHR) and the RXSYNH is updated. RXCHR is set to 1 when the received character is a command, and it is set to 0 if the received character is a data. This mechanism alleviates and simplifies the direct memory access as the character contains its own sync field in the same register.

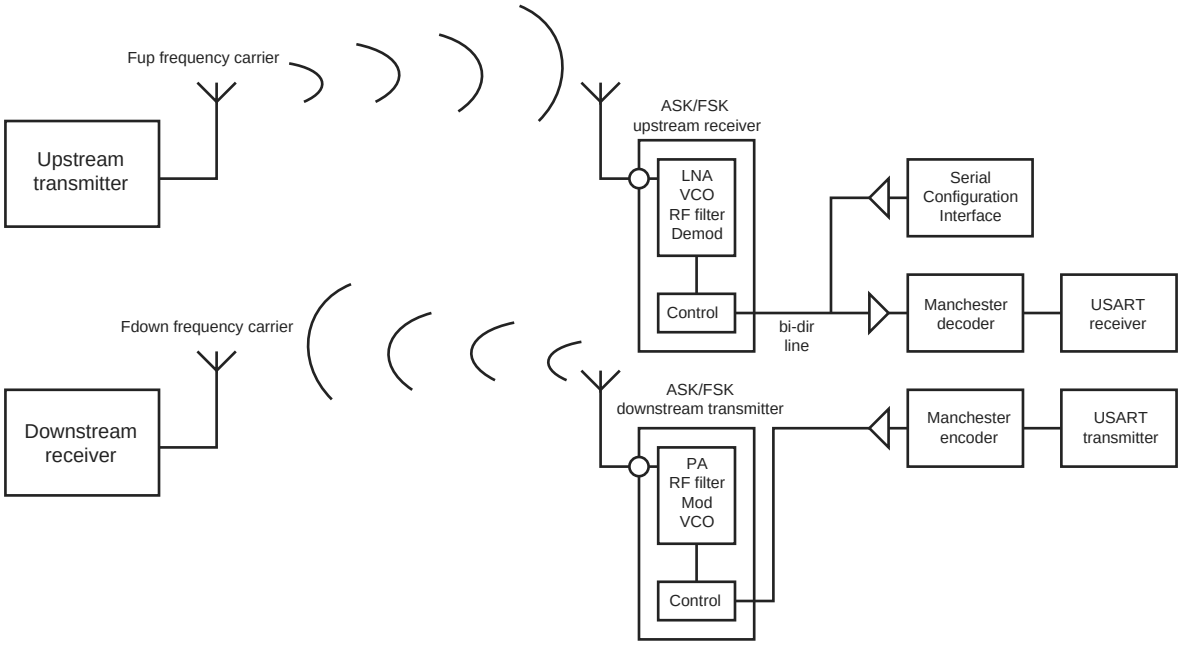
As the decoder is setup to be used in Unipolar mode, the first bit of the frame has to be a zero-to-one transition.

47.7.3.5 Radio Interface: Manchester Encoded USART Application

This section describes low data rate RF transmission systems and their integration with a Manchester encoded USART. These systems are based on transmitter and receiver ICs that support ASK and FSK modulation schemes.

The goal is to perform full-duplex radio transmission of characters using two different frequency carriers. See configuration in Figure 47-16.

Figure 47-16: Manchester Encoded Characters RF Transmission



The USART peripheral is configured as a Manchester encoder/decoder. Looking at the downstream communication channel, Manchester encoded characters are serially sent to the RF transmitter. This may also include a user defined preamble and a start frame delimiter. Mostly, preamble is used in the RF receiver to distinguish between a valid data from a transmitter and signals due to noise. The Manchester stream is then modulated. See Figure 47-17 for an example of ASK modulation scheme. When a logic one is sent to the ASK modulator, the power amplifier, referred to as PA, is enabled and transmits an RF signal at downstream frequency. When a logic zero is transmitted, the RF signal is turned off. If the FSK modulator is activated, two different frequencies are used to transmit data. When a logic 1 is sent, the modulator outputs an RF signal at frequency F0 and switches to F1 if the data sent is a 0. See Figure 47-18.

From the receiver side, another carrier frequency is used. The RF receiver performs a bit check operation examining demodulated data stream. If a valid pattern is detected, the receiver switches to Receiving mode. The demodulated stream is sent to the Manchester decoder. Because of bit checking inside RF IC, the data transferred to the microcontroller is reduced by a user-defined number of bits. The Manchester preamble length is to be defined in accordance with the RF IC configuration.

47.10.31 USART LIN Mode Register

Name: FLEX_US_LINMR

Address: 0xF8034254 (0), 0xF8038254 (1), 0xFC010254 (2), 0xFC014254 (3), 0xFC018254 (4)

Access: Read/Write

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

23	22	21	20	19	18	17	16
–	–	–	–	–	–	SYNCDIS	PDCM

15	14	13	12	11	10	9	8
DLC							

7	6	5	4	3	2	1	0
WKUPTYP	FSDIS	DLM	CHKTYP	CHKDIS	PARDIS	NACT	

This register is relevant only if USART_MODE = 0xA or 0xB in the USART Mode Register.

This register can only be written if the WPEN bit is cleared in the USART Write Protection Mode Register.

NACT: LIN Node Action

Value	Name	Description
0	PUBLISH	The USART transmits the response.
1	SUBSCRIBE	The USART receives the response.
2	IGNORE	The USART does not transmit and does not receive the response.

Values which are not listed in the table must be considered as “reserved”.

PARDIS: Parity Disable

0: In master node configuration, the identifier parity is computed and sent automatically. In master node and slave node configuration, the parity is checked automatically.

1: Whatever the node configuration is, the Identifier parity is not computed/sent and it is not checked.

CHKDIS: Checksum Disable

0: In master node configuration, the checksum is computed and sent automatically. In slave node configuration, the checksum is checked automatically.

1: Whatever the node configuration is, the checksum is not computed/sent and it is not checked.

CHKTYP: Checksum Type

0: LIN 2.0 “enhanced” checksum

1: LIN 1.3 “classic” checksum

DLM: Data Length Mode

0: The response data length is defined by the DLC field of this register.

1: The response data length is defined by the bits 5 and 6 of the identifier (FLEX_US_LINIR.IDCHR).

47.10.43 SPI Control Register

Name: FLEX_SPI_CR

Address: 0xF8034400 (0), 0xF8038400 (1), 0xFC010400 (2), 0xFC014400 (3), 0xFC018400 (4)

Access: Write-only

31	30	29	28	27	26	25	24
FIFODIS	FIFOEN	–	–	–	–	–	LASTXFER
23	22	21	20	19	18	17	16
–	–	–	–	–	–	RXFCLR	TXFCLR
15	14	13	12	11	10	9	8
–	–	–	REQCLR	–	–	–	–
7	6	5	4	3	2	1	0
SWRST	–	–	–	–	–	SPIDIS	SPIEN

SPIEN: SPI Enable

0: No effect.

1: Enables the SPI to transfer and receive data.

SPIDIS: SPI Disable

0: No effect.

1: Disables the SPI.

If a transfer is in progress when SPIDIS is set, the SPI completes the transmission of the shifter register and does not start any new transfer, even if the FLEX_US_THR is loaded.

All pins are set in Input mode after completion of the transmission in progress, if any.

SWRST: SPI Software Reset

0: No effect.

1: Reset the SPI. A software-triggered hardware reset of the SPI interface is performed.

The SPI is in Slave mode after software reset.

REQCLR: Request to Clear the Comparison Trigger

SleepWalking enabled:

0: No effect.

1: Clears the potential clock request currently issued by SPI, thus the potential system wakeup is cancelled.

SleepWalking disabled:

0: No effect.

1: Restarts the comparison trigger to enable FLEX_SPI_RDR loading.

TXFCLR: Transmit FIFO Clear

0: No effect.

1: Empties the Transmit FIFO.

ACMD: Auto CMD Error

Auto CMD12 and Auto CMD23 use this error status. This bit is set to 1 when detecting that one of the 0 to 4 bits in SDMMC_ACESR[4:0] has changed from 0 to 1. In the case of Auto CMD12, this bit is set to 1, not only when errors occur in Auto CMD12, but also when Auto CMD12 is not executed due to the previous command error.

This bit can only be set to 1 if SDMMC_EISTER.ACMD is set to 1. An interrupt can only be generated if SDMMC_EISIER.ACMD is set to 1.

Writing this bit to 1 clears this bit.

0: No error.

1: Error.

ADMA: ADMA Error

This bit is set to 1 when the SDMMC detects errors during an ADMA-based data transfer. The state of the ADMA at an error occurrence is saved in SDMMC_AESR.

In addition, the SDMMC raises this status flag when it detects some invalid description data (Valid = 0) at the ST_FDS state (refer to section “Advanced DMA” in the “SD Host Controller Simplified Specification V3.00”). ADMA Error Status (ERRST) in SDMMC_AESR indicates that an error occured in ST_FDS state. The user may find that the Valid bit is not set at the error descriptor.

This bit can only be set to 1 if SDMMC_EISTER.ADMA is set to 1. An interrupt can only be generated if SDMMC_EISIER.ADMA is set to 1.

Writing this bit to 1 clears this bit.

0: No error.

1: Error.

BOOTAE: Boot Acknowledge Error

This bit is set to 1 when detecting that the e.MMC Boot Acknowledge Status has a value other than “010”.

This bit can only be set to 1 if SDMMC_EISTER.BOOTAE is set to 1. An interrupt can only be generated if SDMMC_EISIER.BOOTAE is set to 1.

Writing this bit to 1 clears this bit.

0: No error.

1: Error.

53.6.45 MCAN Transmit Event FIFO Configuration

Name: MCAN_TXEFC

Address: 0xF80540F0 (0), 0xFC0500F0 (1)

Access: Read/Write

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

This register can only be written if the bits CCE and INIT are set in MCAN CC Control Register.

EFSA: Event FIFO Start Address

Start address of Tx Event FIFO in Message RAM (32-bit word address, see Figure 53-12).

Write EFSA with the bits [15:2] of the 32-bit address.

EFS: Event FIFO Size

0: Tx Event FIFO disabled.

1-32: Number of Tx Event FIFO elements.

>32: Values greater than 32 are interpreted as 32.

The Tx Event FIFO elements are indexed from 0 to EFS - 1.

EFWM: Event FIFO Watermark

0: Watermark interrupt disabled.

1-32: Level for Tx Event FIFO watermark interrupt (MCAN_IR.TEFW).

>32: Watermark interrupt disabled.

54.6.5 Operating Modes

Each channel can operate independently in two different modes:

- Capture mode provides measurement on signals.
- Waveform mode provides wave generation.

The TC operating mode is programmed with TC_CMRx.WAVE.

In Capture mode, TIOAx and TIOBx are configured as inputs.

In Waveform mode, TIOAx is always configured to be an output and TIOBx is an output if it is not selected to be the external trigger.

54.6.6 Trigger

A trigger resets the counter and starts the counter clock. Three types of triggers are common to both modes, and a fourth external trigger is available to each mode.

Regardless of the trigger used, it will be taken into account at the following active edge of the selected clock. This means that the counter value can be read differently from zero just after a trigger, especially when a low frequency signal is selected as the clock.

The following triggers are common to both modes:

- Software Trigger: Each channel has a software trigger, available by setting TC_CCR.SWTRG.
- 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 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 TC_CMRx.CPCTRG is set .

The channel can also be configured to have an external trigger. In Capture mode, the external trigger signal can be selected between TIOAx and TIOBx. In Waveform mode, an external event can be programmed on one of the following signals: TIOBx, XC0, XC1 or XC2. This external event can then be programmed to perform a trigger by setting TC_CMRx.ENETRG.

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

54.6.7 Capture Mode

Capture mode is entered by clearing TC_CMRx.WAVE.

Capture mode allows the TC channel to perform measurements such as pulse timing, frequency, period, duty cycle and phase on TIOAx and TIOBx signals which are considered as inputs.

Figure 54-6 shows the configuration of the TC channel when programmed in Capture mode.

54.6.8 Capture Registers A and B

Registers A and B (TC_RA and TC_RB) are used as capture registers. They can be loaded with the counter value when a programmable event occurs on the signal TIOAx.

TC_CMRx.LDRA defines the TIOAx selected edge for the loading of TC_RA, and TC_CMRx.LDRB defines the TIOAx selected edge for the loading of TC_RB.

The subsampling ratio defined by TC_CMRx.SBSMPLR is applied to these selected edges, so that the loading of Register A and Register B occurs once every 1, 2, 4, 8 or 16 selected edges.

TC_RA is loaded only if it has not been loaded since the last trigger or if TC_RB has been loaded since the last loading of TC_RA.

TC_RB is loaded only if TC_RA has been loaded since the last trigger or the last loading of TC_RB.

Loading TC_RA or TC_RB before the read of the last value loaded sets TC_SR.LOVRS. In this case, the old value is overwritten.

When DMA is used, the Register AB (TC_RAB) address must be configured as source address of the transfer. TC_RAB provides the next unread value from TC_RA and TC_RB. It may be read by the DMA after a request has been triggered upon loading TC_RA or TC_RB.

54.6.9 Transfer with DMAC in Capture Mode

The DMAC can perform access from the TC to system memory in Capture mode only.

Figure 54-5 illustrates how TC_RA and TC_RB can be loaded in the system memory without processor intervention.

3. Generating the Tag using length of *AAD*, length of *C* and J_0 (refer to NIST documentation for details).

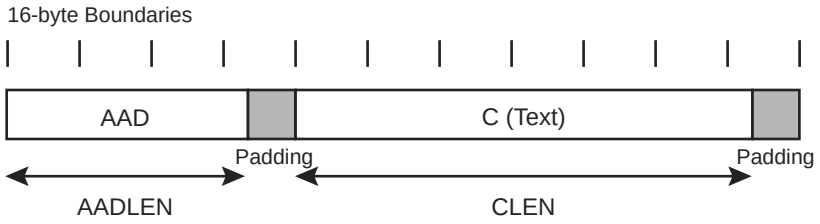
The Tag generation can be done either automatically, after the end of *AAD*/*C* processing if AES_MR.GTAGEN is set, or manually using AES_GHASHRx.GHASH (see subsections Processing a Complete Message with Tag Generation and Manual GCM Tag Generation for details).

- Processing a Complete Message with Tag Generation

Use this procedure only if J_0 four LSB bytes ≠ 0xFFFFFFFF.

NOTE: If J_0 four LSB bytes = 0xFFFFFFFF or if the value is unknown, use the procedure described in Processing a Complete Message without Tag Generation followed by the procedure in Manual GCM Tag Generation.

Figure 60-6: Full Message Alignment



To process a complete message with Tag generation, the sequence is as follows:

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '1'.
2. Set the AES Key Register and wait until AES_ISR.DATRDY is set (GCM hash subkey generation complete); use interrupt if needed. See Section 60.4.6.2 "Key Writing and Automatic Hash Subkey Calculation".
3. Calculate the J_0 value as described in NIST documentation $J_0 = IV \,||\, 0^{31} \,||\, 1$ when len(*IV*) = 96 and $J_0 = \text{GHASH}_H(IV \,||\, 0^{s+64} \,||\, [\text{len}(IV)]_{64})$ if len(*IV*) ≠ 96. See Processing a Message with only AAD (GHASHH) for J_0 generation.
4. Set AES_IVRx.IV with inc32(J_0) (J_0 + 1 on 32 bits).
5. Configure AES_AADLENR.AADLEN and AES_CLENR.CLEN.
6. Fill AES_IDATARx.IDATA with the message to process according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).
7. Wait for TAGRDY to be set (use interrupt if needed), then read AES_TAGRx.TAG to obtain the authentication tag of the message.

- Processing a Complete Message without Tag Generation

Processing a message without generating the Tag can be used to customize the Tag generation, or to process a fragmented message. To manually generate the GCM Tag, see Manual GCM Tag Generation.

To process a complete message without Tag generation, the sequence is as follows:

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '0'.
2. Set the AES Key Register and wait until AES_ISR.DATRDY is set (GCM hash subkey generation complete); use interrupt if needed. After the GCM hash subkey generation is complete the GCM hash subkey can be read or overwritten with specific value in AES_GCMHRx. See Section 60.4.6.2 "Key Writing and Automatic Hash Subkey Calculation".
3. Calculate the J_0 value as described in NIST documentation $J_0 = IV \,||\, 0^{31} \,||\, 1$ when len(*IV*) = 96 and $J_0 = \text{GHASH}_H(IV \,||\, 0^{s+64} \,||\, [\text{len}(IV)]_{64})$ if len(*IV*) ≠ 96. See Processing a Message with only AAD (GHASHH) for J_0 generation example when len(*IV*) ≠ 96.
4. Set AES_IVRx.IV with inc32(J_0) (J_0 + 1 on 32 bits).
5. Configure AES_AADLENR.AADLEN and AES_CLENR.CLEN.
6. Fill AES_IDATARx.IDATA with the message to process according to the SMOD configuration used. If Manual Mode or Auto Mode is used, the DATRDY bit indicates when the data have been processed (however, no output data are generated when processing *AAD*).
7. Make sure the last output data have been read if AES_CLENR.CLEN ≠ 0 (or wait for DATRDY), then read AES_GHASHRx.GHASH to obtain the hash value after the last processed data.

- Processing a Fragmented Message without Tag Generation

If needed, a message can be processed by fragments, in such case automatic GCM Tag generation is not supported.

To process a message by fragments, the sequence is as follows:

- First fragment:

1. Set AES_MR.OPMOD to GCM and AES_MR.GTAGEN to '0'.

65.7.17 ADC Compare Window Register

Name: ADC_CWR

Address: 0xFC030044

Access: Read/Write

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

This register can only be written if the WPEN bit is cleared in the ADC Write Protection Mode Register.

LOWTHRES: Low Threshold

Low threshold associated to compare settings of ADC_EMR.

HIGHTHRES: High Threshold

High threshold associated to compare settings of ADC_EMR.

66.26 GMAC Timings

66.26.1 Timing Conditions

Timings assuming a capacitance load on data and clock are given in Table 66-99.

Table 66-99: Capacitance Load on Data, Clock Pads

Supply	Corner	
	Max	Min
3.3V	20 pF	0 pF

66.26.2 Timing Constraints

Table 66-100: Ethernet MAC Signals Relative to GMDC

Symbol	Parameter	Min	Max	Unit
$EMAC_1$	Setup for GMDIO from GMDC rising	10	–	ns
$EMAC_2$	Hold for GMDIO from GMDC rising	10	–	ns
$EMAC_3$	GMDIO toggling from GMDC rising(1)	0	300	ns

Note 1: For Ethernet MAC output signals, minimum and maximum access time are defined. The minimum access time is the time between the GMDC rising edge and the signal change. The maximum access timing is the time between the GMDC rising edge and the signal stabilizes. Figure 66-41 illustrates minimum and maximum accesses for $EMAC_3$.

Figure 66-41: Minimum and Maximum Access Time of Ethernet MAC Output Signals