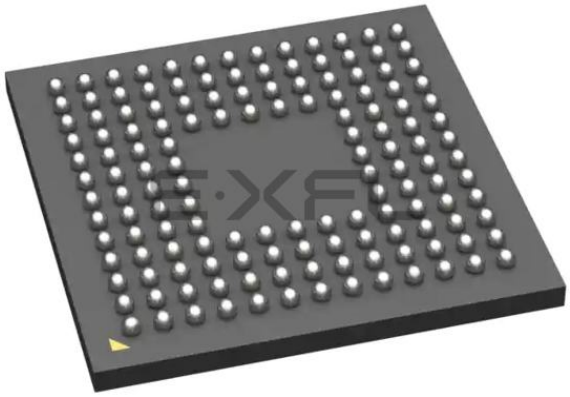


Welcome to E-XFL.COM

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	Active
Core Processor	ARM® Cortex®-M7
Core Size	32-Bit Single-Core
Speed	300MHz
Connectivity	CANbus, EBI/EMI, Ethernet, I²C, IrDA, LINbus, MMC/SD/SDIO, QSPI, SPI, UART/USART, USB
Peripherals	Brown-out Detect/Reset, DMA, I²S, POR, PWM, WDT
Number of I/O	114
Program Memory Size	2MB (2M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 24x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	144-UFBGA
Supplier Device Package	144-UFBGA (6x6)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsame70q21b-cfnt

Since this function is executed from ROM, this allows Flash programming (such as sector write) to be done by code running in Flash.

The IAP function entry point is retrieved by reading the NMI vector in ROM (0x00800008).

This function takes two arguments as parameters:

- the index of the Flash bank to be programmed: 0 for EEFC0, 1 for EEFC1. For devices with only one bank, this parameter has no effect and can be either 0 or 1, only EEFC0 will be accessed.
- the command to be sent to the EEFC Command register.

This function returns the value of the EEFC_FSR register.

An example of IAP software code follows:

```
// Example: How to write data in page 200 of the flash memory using ROM IAP
function

flash_page_num = 200

flash_cmd = 0

flash_status = 0

eefc_index = 0 (0 for EEFC0, 1 for EEFC1)

// Initialize the function pointer (retrieve function address from NMI
vector)*/

iap_function_address = 0x00800008

// Fill the Flash page buffer at address 200 with the data to be written

for i=0, i < page_size, i++ do

flash_sector_200_address[i] = your_data[i]

// Prepare the command to be sent to the EEFC Command register: key, page
number and write command

flash_cmd = (0x5A << 24) | (flash_page_num << 8) | flash_write_command;

// Call the IAP function with the right parameters and retrieve the status in
flash_status after completion

flash_status = iap_function (eefc_index, flash_cmd);
```

27.6.7 RTC Status Register

Name: RTC_SR
Offset: 0x18
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
			TDERR	CALEV	TIMEV	SEC	ALARM	ACKUPD
Access			R	R	R	R	R	R
Reset			0	0	0	0	0	0

Bit 5 – TDERR Time and/or Date Free Running Error

Value	Name	Description
0	CORRECT	The internal free running counters are carrying valid values since the last read of the Status Register (RTC_SR).
1	ERR_TIMEDATE	The internal free running counters have been corrupted (invalid date or time, non-BCD values) since the last read and/or they are still invalid.

Bit 4 – CALEV Calendar Event
The calendar event is selected in the CALEVSEL field in the Control Register (RTC_CR) and can be any one of the following events: week change, month change and year change.

Value	Name	Description
0	NO_CALEVENT	No calendar event has occurred since the last clear.
1	CALEVENT	At least one calendar event has occurred since the last clear.

Bit 3 – TIMEV Time Event
The time event is selected in the TIMEVSEL field in the Control Register (RTC_CR) and can be any one of the following events: minute change, hour change, noon, midnight (day change).

37.6.12 ISI Interrupt Enable Register

Name: ISI_IER
Offset: 0x2C
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
					FR_OVR	CRC_ERR	C_OVR	P_OVR
Access					W	W	W	W
Reset					–	–	–	–

Bit	23	22	21	20	19	18	17	16
							CXFR_DONE	PXFR_DONE
Access							W	W
Reset							–	–

Bit	15	14	13	12	11	10	9	8
						VSYNC		
Access						W		
Reset						–		

Bit	7	6	5	4	3	2	1	0
						SRST	DIS_DONE	
Access						W	W	
Reset						–	–	

Bit 27 – FR_OVR Frame Rate Overflow Interrupt Enable

Value	Description
0	No effect.
1	Enables the corresponding interrupt.

Bit 26 – CRC_ERR Embedded Synchronization CRC Error Interrupt Enable

Value	Description
0	No effect.
1	Enables the corresponding interrupt.

Bit 25 – C_OVR Codec Datapath Overflow Interrupt Enable

Value	Description
0	No effect.
1	Enables the corresponding interrupt.

Bit 24 – P_OVR Preview Datapath Overflow Interrupt Enable

Value	Description
0	No effect.
1	Enables the corresponding interrupt.

has been used. It re-initializes to the receive buffer queue base address if any descriptor has its wrap bit set.

As receive AHB buffers are used, the receive AHB buffer manager sets bit zero of the first word of the descriptor to logic one indicating the AHB buffer has been used.

Software should search through the “used” bits in the AHB buffer descriptors to find out how many frames have been received, checking the start of frame and end of frame bits.

When the DMA is configured in the packet buffer Partial Store And Forward mode, received frames are written out to the AHB buffers as soon as enough frame data exists in the packet buffer. For both cases, this may mean several full AHB buffers are used before some error conditions can be detected. If a receive error is detected the receive buffer currently being written will be recovered. Previous buffers will not be recovered. As an example, when receiving frames with cyclic redundancy check (CRC) errors or excessive length, it is possible that a frame fragment might be stored in a sequence of AHB receive buffers. Software can detect this by looking for start of frame bit set in a buffer following a buffer with no end of frame bit set.

To function properly, a 10/100 Ethernet system should have no excessive length frames or frames greater than 128 Bytes with CRC errors. Collision fragments will be less than 128 Bytes long, therefore it will be a rare occurrence to find a frame fragment in a receive AHB buffer, when using the default value of 128 Bytes for the receive buffers size.

When in packet buffer full store and forward mode, only good received frames are written out of the DMA, so no fragments will exist in the AHB buffers due to MAC receiver errors. There is still the possibility of fragments due to DMA errors, for example used bit read on the second buffer of a multi-buffer frame.

If bit zero of the receive buffer descriptor is already set when the receive buffer manager reads the location of the receive AHB buffer, the buffer has been already used and cannot be used again until software has processed the frame and cleared bit zero. In this case, the “buffer not available” bit in the receive status register is set and an interrupt triggered. The receive resource error statistics register is also incremented.

When the DMA is configured in the packet buffer full store and forward mode, the user can optionally select whether received frames should be automatically discarded when no AHB buffer resource is available. This feature is selected via the DMA Discard Receive Packets bit in the DMA Configuration register (GMAC_DCFGR.DDRP). By default, the received frames are not automatically discarded. If this feature is off, then received packets will remain to be stored in the SRAM-based packet buffer until AHB buffer resource next becomes available. This may lead to an eventual packet buffer overflow if packets continue to be received when bit zero (used bit) of the receive buffer descriptor remains set.
Note: After a used bit has been read, the receive buffer manager will re-read the location of the receive buffer descriptor every time a new packet is received. When the DMA is not configured in the packet buffer full store and forward mode and a used bit is read, the frame currently being received will be automatically discarded.

When the DMA is configured in the packet buffer full store and forward mode, a receive overrun condition occurs when the receive SRAM-based packet buffer is full, or because HRESP was not OK. In all other modes, a receive overrun condition occurs when either the AHB bus was not granted quickly enough, or because HRESP was not OK, or because a new frame has been detected by the receive block, but the status update or write back for the previous frame has not yet finished. For a receive overrun condition, the receive overrun interrupt is asserted and the buffer currently being written is recovered. The next frame that is received whose address is recognized reuses the buffer.

38.8.3 GMAC Network Status Register

Name: GMAC_NSR
Offset: 0x008
Reset: 0x000001X0
Property: Read-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
						IDLE	MDIO	
Access						R	R	
Reset						0	0	

Bit 2 – IDLE PHY Management Logic Idle
The PHY management logic is idle (i.e., has completed).

Bit 1 – MDIO MDIO Input Status
Returns status of the MDIO pin.

38.8.83 GMAC 1588 Timer Increment Sub-nanoseconds Register

Name: GMAC_TISUBN
Offset: 0x1BC
Reset: 0x00000000
Property: -

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	LSBTIR[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	LSBTIR[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 15:0 – LSBTIR[15:0] Lower Significant Bits of Timer Increment Register
Lower significant bits of Timer Increment Register [15:0], giving a 24-bit timer_increment counter. These bits are the sub-ns value which the 1588 timer will be incremented each clock cycle. Bit n = $2^{(n-16)}$ ns giving a resolution of approximately 15.2E^{-15} sec.

Offset	Name	Bit Pos.								
0x01D4	USBHS_DEVEPTIMR5 (ISOENPT)	7:0	SHORTPACKETE	CRCERRE	OVERFE	HBISOFLUSHE	HBISOINERRE	UNDERFE	RXOUTE	TXINE
		15:8		FIFOCON	KILLBK	NBUSYBKE		ERRORTRANSE	DATAXE	MDATAE
		23:16						RSTDT		EPDISHDMA
		31:24								
0x01D8	USBHS_DEVEPTIMR6	7:0	SHORTPACKETE	STALLEDE	OVERFE	NAKINE	NAKOUTE	RXSTPE	RXOUTE	TXINE
		15:8		FIFOCON	KILLBK	NBUSYBKE				
		23:16					STALLRQ	RSTDT	NYETDIS	EPDISHDMA
		31:24								
0x01D8	USBHS_DEVEPTIMR6 (ISOENPT)	7:0	SHORTPACKETE	CRCERRE	OVERFE	HBISOFLUSHE	HBISOINERRE	UNDERFE	RXOUTE	TXINE
		15:8		FIFOCON	KILLBK	NBUSYBKE		ERRORTRANSE	DATAXE	MDATAE
		23:16						RSTDT		EPDISHDMA
		31:24								
0x01DC	USBHS_DEVEPTIMR7	7:0	SHORTPACKETE	STALLEDE	OVERFE	NAKINE	NAKOUTE	RXSTPE	RXOUTE	TXINE
		15:8		FIFOCON	KILLBK	NBUSYBKE				
		23:16					STALLRQ	RSTDT	NYETDIS	EPDISHDMA
		31:24								
0x01DC	USBHS_DEVEPTIMR7 (ISOENPT)	7:0	SHORTPACKETE	CRCERRE	OVERFE	HBISOFLUSHE	HBISOINERRE	UNDERFE	RXOUTE	TXINE
		15:8		FIFOCON	KILLBK	NBUSYBKE		ERRORTRANSE	DATAXE	MDATAE
		23:16						RSTDT		EPDISHDMA
		31:24								
0x01E0	USBHS_DEVEPTIMR8	7:0	SHORTPACKETE	STALLEDE	OVERFE	NAKINE	NAKOUTE	RXSTPE	RXOUTE	TXINE
		15:8		FIFOCON	KILLBK	NBUSYBKE				
		23:16					STALLRQ	RSTDT	NYETDIS	EPDISHDMA
		31:24								
0x01E0	USBHS_DEVEPTIMR8 (ISOENPT)	7:0	SHORTPACKETE	CRCERRE	OVERFE	HBISOFLUSHE	HBISOINERRE	UNDERFE	RXOUTE	TXINE
		15:8		FIFOCON	KILLBK	NBUSYBKE		ERRORTRANSE	DATAXE	MDATAE
		23:16						RSTDT		EPDISHDMA
		31:24								
0x01E4	USBHS_DEVEPTIMR9	7:0	SHORTPACKETE	STALLEDE	OVERFE	NAKINE	NAKOUTE	RXSTPE	RXOUTE	TXINE
		15:8		FIFOCON	KILLBK	NBUSYBKE				
		23:16					STALLRQ	RSTDT	NYETDIS	EPDISHDMA
		31:24								
0x01E4	USBHS_DEVEPTIMR9 (ISOENPT)	7:0	SHORTPACKETE	CRCERRE	OVERFE	HBISOFLUSHE	HBISOINERRE	UNDERFE	RXOUTE	TXINE

Offset	Name	Bit Pos.								
0x0308	USBHS_DEVDMACONTROL1	7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB
		15:8								
		23:16	BUFF_LENGTH[7:0]							
		31:24	BUFF_LENGTH[15:8]							
0x030C	USBHS_DEVDMASTATUS1	7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB
		15:8								
		23:16	BUFF_COUNT[7:0]							
		31:24	BUFF_COUNT[15:8]							
0x0310	USBHS_DEVDMANXTDSC2	7:0	NXT_DSC_ADD[7:0]							
		15:8	NXT_DSC_ADD[15:8]							
		23:16	NXT_DSC_ADD[23:16]							
		31:24	NXT_DSC_ADD[31:24]							
0x0314	USBHS_DEVDMAADDRESS2	7:0	BUFF_ADD[7:0]							
		15:8	BUFF_ADD[15:8]							
		23:16	BUFF_ADD[23:16]							
		31:24	BUFF_ADD[31:24]							
0x0318	USBHS_DEVDMACONTROL2	7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB
		15:8								
		23:16	BUFF_LENGTH[7:0]							
		31:24	BUFF_LENGTH[15:8]							
0x031C	USBHS_DEVDMASTATUS2	7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB
		15:8								
		23:16	BUFF_COUNT[7:0]							
		31:24	BUFF_COUNT[15:8]							
0x0320	USBHS_DEVDMANXTDSC3	7:0	NXT_DSC_ADD[7:0]							
		15:8	NXT_DSC_ADD[15:8]							
		23:16	NXT_DSC_ADD[23:16]							
		31:24	NXT_DSC_ADD[31:24]							
0x0324	USBHS_DEVDMAADDRESS3	7:0	BUFF_ADD[7:0]							
		15:8	BUFF_ADD[15:8]							
		23:16	BUFF_ADD[23:16]							
		31:24	BUFF_ADD[31:24]							
0x0328	USBHS_DEVDMACONTROL3	7:0	BURST_LCK	DESC_LD_IT	END_BUFFIT	END_TR_IT	END_B_EN	END_TR_EN	LDNXT_DSC	CHANN_ENB
		15:8								
		23:16	BUFF_LENGTH[7:0]							
		31:24	BUFF_LENGTH[15:8]							
0x032C	USBHS_DEVDMASTATUS3	7:0		DESC_LDST	END_BF_ST	END_TR_ST			CHANN_ACT	CHANN_ENB
		15:8								
		23:16	BUFF_COUNT[7:0]							
		31:24	BUFF_COUNT[15:8]							
0x0330	USBHS_DEVDMANXTDSC4	7:0	NXT_DSC_ADD[7:0]							
		15:8	NXT_DSC_ADD[15:8]							
		23:16	NXT_DSC_ADD[23:16]							
		31:24	NXT_DSC_ADD[31:24]							
0x0334	USBHS_DEVDMAADDRESS4	7:0	BUFF_ADD[7:0]							
		15:8	BUFF_ADD[15:8]							

42.7.11 QSPI Instruction Code Register

Name: QSPI_ICR
Offset: 0x34
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	OPT[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	INST[7:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 23:16 – OPT[7:0] Option Code
Option code to send to the serial Flash memory.

Bits 7:0 – INST[7:0] Instruction Code
Instruction code to send to the serial Flash memory.

43.7.10 TWIHS Interrupt Disable Register

Name: TWIHS_IDR
Offset: 0x28
Reset: –
Property: Write-only

The following configuration values are valid for all listed bit names of this register:

0: No effect.

1: Disables the corresponding interrupt.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
			SMBHHM	SMBDAM	PECERR	TOUT		MCACK
Access			W	W	W	W		W
Reset			–	–	–	–		–

Bit	15	14	13	12	11	10	9	8
					EOSACC	SCL_WS	ARBLST	NACK
Access					W	W	W	W
Reset					–	–	–	–

Bit	7	6	5	4	3	2	1	0
	UNRE	OVRE	GACC	SVACC		TXRDY	RXRDY	TXCOMP
Access	W	W	W	W		W	W	W
Reset	–	–	–	–		–	–	–

Bit 21 – SMBHHM SMBus Host Header Address Match Interrupt Disable

Bit 20 – SMBDAM SMBus Default Address Match Interrupt Disable

Bit 19 – PECERR PEC Error Interrupt Disable

Bit 18 – TOUT Timeout Error Interrupt Disable

Bit 16 – MCACK Master Code Acknowledge Interrupt Disable

Bit 11 – EOSACC End Of Slave Access Interrupt Disable

Bit 10 – SCL_WS Clock Wait State Interrupt Disable

Bit 9 – ARBLST Arbitration Lost Interrupt Disable

Bit 8 – NACK Not Acknowledge Interrupt Disable

43.7.14 TWIHS Transmit Holding Register

Name: TWIHS_THR
Offset: 0x34
Reset: 0x00000000
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
	TXDATA[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – TXDATA[7:0] Master or Slave Transmit Holding Data

49.5.7.3 Tx Buffer Element

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO / Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO / Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler distinguishes between dedicated Tx Buffers and Tx FIFO / Tx Queue by evaluating the Tx Buffer configuration TXBC.TFQS and TXBC.NDTB. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via register TXESC.

Table 49-8. Tx Buffer Element

	31			24	23				16	15	8	7	0
T0	ESI	XTD	RTR	ID[28:0]									
T1	MM[7:0]				EFC	reserved	FDF	BRS	DLC[3:0]	reserved			
T2	DB3[7:0]				DB2[7:0]					DB1[7:0]		DB0[7:0]	
T3	DB7[7:0]				DB6[7:0]					DB5[7:0]		DB4[7:0]	
...	...				...					...		...	
Tn	DBm[7:0]				DBm-1[7:0]					DBm-2[7:0]		DBm-3[7:0]	

• T0 Bit 30 ESI: Error State Indicator

T0 Bit 31 ESI: Error State Indicator

0: ESI bit in CAN FD format depends only on error passive flag

1: ESI bit in CAN FD format transmitted recessive

Note: The ESI bit of the transmit buffer is or’ed with the error passive flag to decide the value of the ESI bit in the transmitted FD frame. As required by the CAN FD protocol specification, an error active node may optionally transmit the ESI bit recessive, but an error passive node will always transmit the ESI bit recessive. This feature can be used in gateway applications when a message from an error passive node is routed to another CAN network.

• T0 Bit 30 XTD: Extended Identifier

0: 11-bit standard identifier.

1: 29-bit extended identifier.

• T0 Bit 29 RTR: Remote Transmission Request

0: Transmit data frame.

1: Transmit remote frame.

Note: When RTR = 1, the MCAN transmits a remote frame according to ISO11898-1, even if MCAN_CCCR.FDOE enables the transmission in CAN FD format.

• T0 Bits 28:0 ID[28:0]: Identifier

Standard or extended identifier depending on bit XTD. A standard identifier has to be written to ID[28:18].

• T1 Bits 31:24 MM[7:0]: Message Marker

Written by processor during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status.

49.6.25 MCAN New Data 1

Name: MCAN_NDAT1
Offset: 0x98
Reset: 0x00000000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Bits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31 – NDx New Data
The register holds the New Data flags of Receive Buffers 0 to 31. The flags are set when the respective Receive Buffer has been updated from a received frame. The flags remain set until the processor clears them. A flag is cleared by writing a ‘1’ to the corresponding bit position. Writing a ‘0’ has no effect. A hard reset will clear the register.

Value	Description
0	Receive Buffer not updated
1	Receive Buffer updated from new message

50.7 Register Summary

Note: The register TC_CMR has two modes, Capture Mode and Waveform Mode. In this register summary, both modes are displayed

Offset	Name	Bit Pos.								
0x00	TC_CCR0	7:0						SWTRG	CLKDIS	CLKEN
		15:8								
		23:16								
		31:24								
0x04	TC_CMR0	7:0	LDBDIS	LDBSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
		15:8	WAVE	CPCTRG				ABETRG	ETRGEDG[1:0]	
		23:16		SBSMPLR[2:0]			LDRB[1:0]		LDRA[1:0]	
		31:24								
0x04	TC_CMR0	7:0	CPCDIS	CPCSTOP	BURST[1:0]		CLKI	TCCLKS[2:0]		
		15:8	WAVE	WAVSEL[1:0]		ENETRG	EEVT[1:0]		EEVTEDG[1:0]	
		23:16	ASWTRG[1:0]		AEEVT[1:0]		ACPC[1:0]		ACPA[1:0]	
		31:24	BSWTRG[1:0]		BEEVT[1:0]		BCPC[1:0]		BCPB[1:0]	
0x08	TC_SMMR0	7:0							DOWN	GCEN
		15:8								
		23:16								
		31:24								
0x0C	TC_RAB0	7:0	RAB[7:0]							
		15:8	RAB[15:8]							
		23:16	RAB[23:16]							
		31:24	RAB[31:24]							
0x10	TC_CV0	7:0	CV[7:0]							
		15:8	CV[15:8]							
		23:16	CV[23:16]							
		31:24	CV[31:24]							
0x14	TC_RA0	7:0	RA[7:0]							
		15:8	RA[15:8]							
		23:16	RA[23:16]							
		31:24	RA[31:24]							
0x18	TC_RB0	7:0	RB[7:0]							
		15:8	RB[15:8]							
		23:16	RB[23:16]							
		31:24	RB[31:24]							
0x1C	TC_RC0	7:0	RC[7:0]							
		15:8	RC[15:8]							
		23:16	RC[23:16]							
		31:24	RC[31:24]							
0x20	TC_SR0	7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS
		15:8								
		23:16						MTIOB	MTIOA	CLKSTA
		31:24								
0x24	TC_IER0	7:0	ETRGS	LDRBS	LDRAS	CPCS	CPBS	CPAS	LOVRS	COVFS

51.7.2 PWM Enable Register

Name: PWM_ENA
Offset: 0x04
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
					CHID3	CHID2	CHID1	CHID0
Access					W	W	W	W
Reset					0	0	0	–

Bits 0, 1, 2, 3 – CHIDx Channel ID

Value	Description
0	No effect.
1	Enable PWM output for channel x.

Value	Description
0	The SW write protection x of the register group x is disabled.
1	The SW write protection x of the register group x is enabled.

55.6.2 ICM Control Register

Name: ICM_CTRL
Offset: 0x04
Reset: –
Property: Write-only

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
	RMEN[3:0]				RMDIS[3:0]			
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	–	0	0	0	–

Bit	7	6	5	4	3	2	1	0
	REHASH[3:0]					SWRST	DISABLE	ENABLE
Access	W	W	W	W		W	W	W
Reset	0	0	0	–		–	–	–

Bits 15:12 – RMEN[3:0] Region Monitoring Enable
Monitoring is activated by default.

Value	Description
0	No effect
1	When bit RMEN[i] is set to one, the monitoring of region with identifier i is activated.

Bits 11:8 – RMDIS[3:0] Region Monitoring Disable

Value	Description
0	No effect
1	When bit RMDIS[i] is set to one, the monitoring of region with identifier i is disabled.

Bits 7:4 – REHASH[3:0] Recompute Internal Hash

Value	Description
0	No effect
1	When REHASH[i] is set to one, Region i digest is re-computed. This bit is only available when region monitoring is disabled.

Bit 2 – SWRST Software Reset

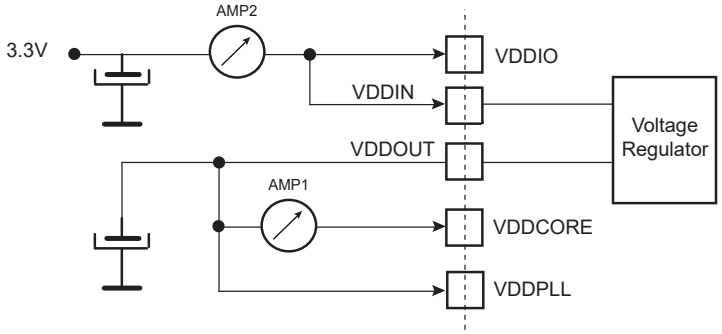
58.3.2 Sleep Mode Current Consumption and Wakeup Time

The Sleep mode configuration and measurements are defined as follows:

- Core clock OFF
- $V_{DDIO} = V_{DDIN} = 3.3V$
- Master Clock (MCK) running at various frequencies with PLLA or the fast RC oscillator
- Fast startup through WKUP0–13 pins
- Current measurement as shown below and associated wakeup time(1)
- All peripheral clocks deactivated
- $T_A = 25°C$

  Note: 1. Wakeup time is defined as the delay between the WKUP event and the execution of the first instruction.

Figure 58-6. Measurement Setup for Sleep Mode



The following tables give current consumption and wakeup time in Sleep mode.

Table 58-13. Typical Sleep Mode Current Consumption vs. Master Clock (MCK) Variation with PLLA

Core Clock/MCK (MHz)	VDDCORE Consumption (AMP1)	Total Consumption (AMP2)	Unit	Wakeup Time	Unit
300/150	20	24	mA	0.85	µs
250/125	17	20		1.05	
150/150	20	24		0.9	
96/96	12.5	15		1.4	
96/48	7.5	10		2.5	
48/48	7	9.5		2.8	
24/24	3.5	5		5.6	
24/12	2	3		10	
12/12	2	3		11.2	
8/8	1.5	2		16.8	
4/4	1.0	1.5		32.9	
4/2	0.9	1		60	
4/1	0.8	1		112.6	

Date	Changes
	Section 18.12.10 “Write Protection Status Register”: in WPVS bit description, replaced two instances of “since the last read of the MATRIX_WPSR” with “since the last write of the MATRIX_WPMR”.
	Section 21. “Enhanced Embedded Flash Controller (EEFC)” Section 21.4.3.2 “Write Commands”: added information on DMA write accesses.
	Section 30. “Power Management Controller (PMC)” Section 30.9 “Asynchronous Partial Wake-up”: inserted new sub-section “Asynchronous Partial Wake-up in Wait Mode (SleepWalking)” to better describe SleepWalking. Section 30.10 “Free-Running Processor Clock”: removed reference to MCK.
	Section 31. “Parallel Input/Output Controller (PIO)” Section 31.2 “Embedded Characteristics”: added bullet on Programmable I/O Drive. Added Section 31.5.12 “Programmable I/O Drive”. Section 31.5.15.4 “Programming Sequence”: “With DMA”: in fifth step, replaced reference to BTCx with ‘DMA status flag to indicate that the buffer transfer is complete’ Table 31-5 “Register Mapping”: added PIO_DRIVER register at offset 0x0118 and added Section 31.6.49 “PIO I/O Drive Register”.
	Section 35. “DMA Controller (XDMAC)” Added Section 35.3 “DMA Controller Peripheral Connections”.
	Section 37. “USB High-Speed Interface (USBHS)” Table 37-1 “Description of USB Pipes/Endpoints“; corrected data in columns ‘DMA’ and ‘High Bandwidth’. Modified signal names to HSDM/DM and HSDP/DP in Figure 37-1 “USBHS Block Diagram” and Table 37-2 “Signal Description”. Updated descriptions. Removed Section 37.3.1 “Application Block Diagram” and Figures 37-2, 37-3 and 37-4. Removed Section 37.4.1 “I/O Lines”. Modified Section 37.5.3.3 “Device Detection”. Section 37.6.2 “General Status Register”, Section 37.6.3 “General Status Clear Register”, Section 37.6.4 “General Status Set Register”: removed bit VBUSRQ and bit description. Bit 9 now reserved in these registers.
24-Feb-15	Section 38. “Ethernet MAC (GMAC)” Section 38.8.13 “GMAC Interrupt Mask Register”: corrected general bit description (swapped definitions provided for 0: and 1:)
	Section 40. “Quad SPI Interface (QSPI)” Section 40.5.4 “Direct Memory Access Controller (DMA)”: added Note on 32-bit aligned DMA write accesses. Figure 40-9 “Instruction Transmission Flow Diagram”: modified text if TFRTYP = 0