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, 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	75
Program Memory Size	1MB (1M 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 10x12b; D/A 2x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	100-TFBGA
Supplier Device Package	100-TFBGA (9x9)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsame70n20a-cnt

If the Watchdog Timer value is greater than WDD, setting bit WDT_CR.WDRSTT causes a watchdog error.

Bit 15 – WDDIS Watchdog Disable
When setting the WDDIS bit, and while it is set, the fields WDV and WDD must not be modified.

Value	Description
0	Enables the Watchdog Timer.
1	Disables the Watchdog Timer.

Bit 13 – WDRSTEN Watchdog Reset Enable

Value	Description
0	A watchdog fault (underflow or error) has no effect on the resets.
1	A watchdog fault (underflow or error) triggers a watchdog reset.

Bit 12 – WDFIEN Watchdog Fault Interrupt Enable

Value	Description
0	A watchdog fault (underflow or error) has no effect on interrupt.
1	A watchdog fault (underflow or error) asserts interrupt.

Bits 11:0 – WDV[11:0] Watchdog Counter Value
Defines the value loaded in the 12-bit watchdog counter.

Table 32-3. Programming Example

Register	Value to be Written
PIO_PER	0x0000_FFFF
PIO_PDR	0xFFFF_0000
PIO_OER	0x0000_00FF
PIO_ODR	0xFFFF_FF00
PIO_IFER	0x0000_0F00
PIO_IFDR	0xFFFF_F0FF
PIO_SODR	0x0000_0000
PIO_CODR	0x0FFF_FFFF
PIO_IER	0x0F00_0F00
PIO_IDR	0xF0FF_F0FF
PIO_MDER	0x0000_000F
PIO_MDDR	0xFFFF_FFF0
PIO_PUDR	0xFFF0_00F0
PIO_PUER	0x000F_FF0F
PIO_PPDDR	0xFF0F_FFFF
PIO_PPDER	0x00F0_0000
PIO_ABCDSR1	0xF0F0_0000
PIO_ABCDSR2	0xFF00_0000
PIO_OWER	0x0000_000F
PIO_OWDR	0x0FFF_FFF0

32.5.16 Register Write Protection

To prevent any single software error from corrupting PIO behavior, certain registers in the address space can be write-protected by setting the WPEN bit in the PIO Write Protection Mode Register (PIO_WPMR).

If a write access to a write-protected register is detected, the WPVS flag in the PIO Write Protection Status Register (PIO_WPSR) is set and the field WPVSRC indicates the register in which the write access has been attempted.

The WPVS bit is automatically cleared after reading the PIO_WPSR.

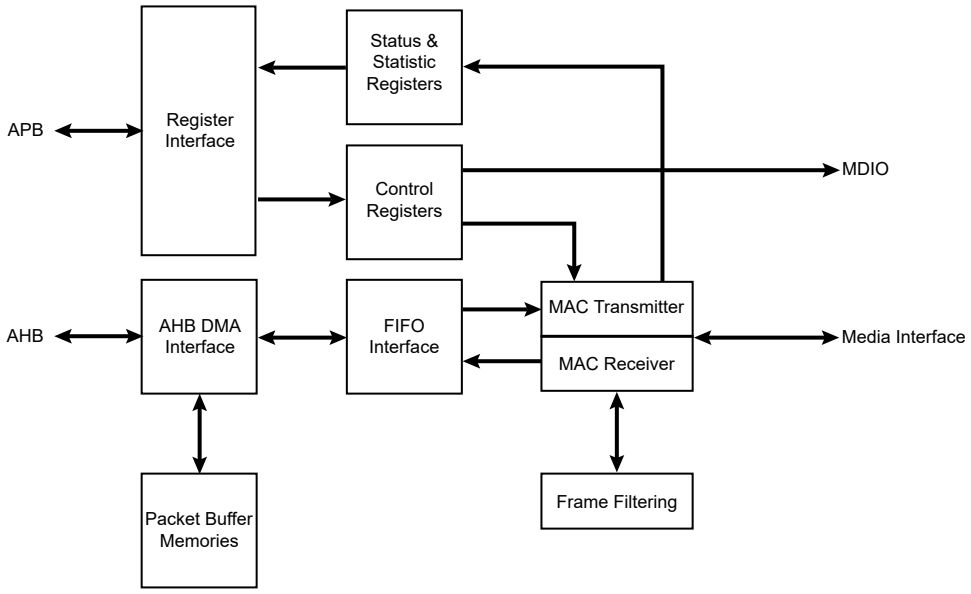
The following registers can be write-protected:

- PIO Enable Register
- PIO Disable Register
- PIO Output Enable Register
- PIO Output Disable Register
- PIO Input Filter Enable Register

Signals: EBI_	Power supply	Pins of the Interfaced Device	
		SDR/LPSDR	NAND Flash
Controller		SDRAMC	NFC
A2–A10	VDDIO	A[0:8]	–
A11	VDDIO	A9	–
SDA10	VDDIO	A10	–
A12	VDDIO	–	–
A13–A14	VDDIO	A[11:12]	–
A15	VDDIO	A13	–
A16/BA0	VDDIO	BA0	–
A17/BA1	VDDIO	BA1	–
A18	VDDIO	–	–
A19	VDDIO	–	–
A20	VDDIO	–	–
A21/NANDALE	VDDIO	–	ALE
A22/NANDCLE	VDDIO	–	CLE
A23	VDDIO	–	–
NCS0	VDDIO	–	–
NCS1/SDCS	VDDIO	SDCS	–
NCS2	VDDIO	–	–
NCS3/NANDCS	VDDIO	–	CE
NANDOE	VDDIO	–	OE
NANDWE	VDDIO	–	WE
NRD	VDDIO	–	–
NWR0/NWE	VDDIO	–	–
NWR1/NBS1	VDDIO	DQM1	–
SDCK	VDDIO	CK	–
SDCKE	VDDIO	CKE	–
RAS	VDDIO	RAS	–
CAS	VDDIO	CAS	–
SDWE	VDDIO	WE	–
Pxx	VDDIO	–	CE
Pxx	VDDIO	–	RDY

38.3 Block Diagram

Figure 38-1. Block Diagram



38.4 Signal Interface

The GMAC includes the following signal interfaces:

- MII, RMII to an external PHY
- MDIO interface for external PHY management
- Slave APB interface for accessing GMAC registers
- Master AHB interface for memory access
- GTSUCOMP signal for TSU timer count value comparison

Table 38-1. GMAC Connections in Different Modes

Signal Name	Function	MII	RMII
GTXCK[(1)]	Transmit Clock or Reference Clock	TXCK	REFCK
GTXEN	Transmit Enable	TXEN	TXEN
GTX[3..0]	Transmit Data	TXD[3:0]	TXD[1:0]
GTXER	Transmit Coding Error	TXER	Not Used
GRXCK	Receive Clock	RXCK	Not Used
GRXDV	Receive Data Valid	RXDV	CRSDV
GRX[3..0]	Receive Data	RXD[3:0]	RXD[1:0]
GRXER	Receive Error	RXER	RXER
GCRS	Carrier Sense and Data Valid	CRS	Not Used
GCOL	Collision Detect	COL	Not Used

will only begin to forward the packet to the AHB when enough packet data is stored in the packet buffer. The amount of packet data required to activate the forwarding process is programmable via watermark registers. These registers are located at the same address as the partial store and forward enable bits.
Note: The minimum operational value for the TX partial store and forward watermark is 20. There is no operational limit for the RX partial store and forward watermark.

Enabling Partial Store and Forward is a useful means to reduce latency, but there are performance implications. The GMAC DMA uses separate transmit and receive lists of buffer descriptors, with each descriptor describing a buffer area in memory. This allows Ethernet packets to be broken up and scattered around the AHB memory space.

38.6.3.3 Receive AHB Buffers

Received frames, optionally including FCS, are written to receive AHB buffers stored in memory. The receive buffer depth is programmable in the range of 64 Bytes to 16 KBytes through the DMA Configuration register (GMAC_DCFGR), with the default being 128 Bytes.

The start location for each receive AHB buffer is stored in memory in a list of receive buffer descriptors at an address location pointed to by the receive buffer queue pointer. The base address for the receive buffer queue pointer is configured in software using the Receive Buffer Queue Base Address register (GMAC_RBQB).

Each list entry consists of two words. The first is the address of the receive AHB buffer and the second the receive status.

If the length of a receive frame exceeds the AHB buffer length, the status word for the used buffer is written with zeroes except for the “Start of Frame” bit, which is always set for the first buffer in a frame.

Bit zero of the address field is written to 1 to show that the buffer has been used. The receive buffer manager then reads the location of the next receive AHB buffer and fills that with the next part of the received frame data. AHB buffers are filled until the frame is complete and the final buffer descriptor status word contains the complete frame status. See the following table for details of the receive buffer descriptor list.

Table 38-2. Receive Buffer Descriptor Entry

Bit	Function
Word 0	
31:2	Address of beginning of buffer
1	Wrap—marks last descriptor in receive buffer descriptor list.
0	Ownership—needs to be zero for the GMAC to write data to the receive buffer. The GMAC sets this to one once it has successfully written a frame to memory. Software has to clear this bit before the buffer can be used again.
Word 1	
31	Global all ones broadcast address detected
30	Multicast hash match
29	Unicast hash match
28	–

To change loop back mode, the following sequence of operations must be followed:

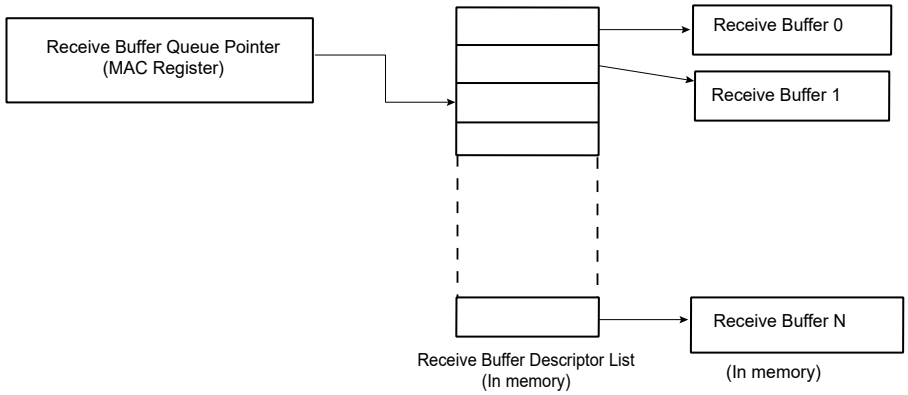
1. Write to Network Control register to disable transmit and receive circuits.
2. Write to Network Control register to change loop back mode.
3. Write to Network Control register to re-enable transmit or receive circuits.
   Note: These writes to the Network Control register cannot be combined in any way.

38.7.1.2 Receive Buffer List

Receive data is written to areas of data (i.e., buffers) in system memory. These buffers are listed in another data structure that also resides in main memory. This data structure (receive buffer queue) is a sequence of descriptor entries as defined in the table Receive Buffer Descriptor Entry.

The Receive Buffer Queue Pointer register points to this data structure.

Figure 38-3. Receive Buffer List



To create the list of buffers:

1. Allocate a number (N) of buffers of X bytes in system memory, where X is the DMA buffer length programmed in the DMA Configuration register.
2. Allocate an area 8N bytes for the receive buffer descriptor list in system memory and create N entries in this list. Mark all entries in this list as owned by GMAC, i.e., bit 0 of word 0 set to 0.
3. Mark the last descriptor in the queue with the wrap bit (bit 1 in word 0 set to 1).
4. Write address of receive buffer descriptor list and control information to GMAC register receive buffer queue pointer
5. The receive circuits can then be enabled by writing to the address recognition registers and the Network Control register.

Note: The queue pointers must be initialized and point to USED descriptors for all queues including those not intended for use.

38.7.1.3 Transmit Buffer List

Transmit data is read from areas of data (the buffers) in system memory. These buffers are listed in another data structure that also resides in main memory. This data structure (Transmit Buffer Queue) is a sequence of descriptor entries as defined in the table Transmit Buffer Descriptor Entry.

The Transmit Buffer Queue Pointer register points to this data structure.

To create this list of buffers:

1. Allocate a number (N) of buffers of between 1 and 2047 bytes of data to be transmitted in system memory. Up to 128 buffers per frame are allowed.

Bit 16 – RWALL Read/Write Allowed
For an OUT pipe, this bit is set when the current bank is not full, i.e., the software can write further data into the FIFO.

For an IN pipe, this bit is set when the current bank is not empty, i.e., the software can read further data from the FIFO.

This bit is cleared otherwise.

This bit is also cleared when the RXSTALLDI or the PERRI bit = 1.

Bits 15:14 – CURRBK[1:0] Current Bank
For a non-control pipe, this field indicates the number of the current bank.

This field may be updated 1 clock cycle after the RWALL bit changes, so the user should not poll it as an interrupt bit.

Value	Name	Description
0	BANK0	Current bank is bank0
1	BANK1	Current bank is bank1
2	BANK2	Current bank is bank2
3	Reserved	

Bits 13:12 – NBUSYBK[1:0] Number of Busy Banks
This field indicates the number of busy banks.

For an OUT pipe, this field indicates the number of busy banks, filled by the user, ready for an OUT transfer. When all banks are busy, this triggers a PEP_x interrupt if USBHS_HSTPIPIMRx.NBUSYBKE = 1.

For an IN pipe, this field indicates the number of busy banks filled by IN transaction from the device. When all banks are free, this triggers a PEP_x interrupt if USBHS_HSTPIPIMRx.NBUSYBKE = 1.

Value	Name	Description
0	0_BUSY	0 busy bank (all banks free)
1	1_BUSY	1 busy bank
2	2_BUSY	2 busy banks
3	3_BUSY	3 busy banks

Bits 9:8 – DTSEQ[1:0] Data Toggle Sequence
This field indicates the data PID of the current bank.

For an OUT pipe, this field indicates the data toggle of the next packet that is to be sent.

For an IN pipe, this field indicates the data toggle of the received packet stored in the current bank.

Value	Name	Description
0	DATA0	Data0 toggle sequence
1	DATA1	Data1 toggle sequence
2	Reserved	
3	Reserved	

Bit 7 – SHORTPACKETI Short Packet Interrupt

Bit 5 – OVERFIEC Overflow Interrupt Disable

Bit 4 – NAKEDEC NAKed Interrupt Disable

Bit 3 – PERREC Pipe Error Interrupt Disable

Bit 2 – UNDERFIEC Underflow Interrupt Disable

Bit 1 – TXOUTEC Transmitted OUT Data Interrupt Disable

Bit 0 – RXINEC Received IN Data Interrupt Disable

43.7.12 TWIHS Receive Holding Register

Name: TWIHS_RHR
Offset: 0x30
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
	RXDATA[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bits 7:0 – RXDATA[7:0] Master or Slave Receive Holding Data

43.7.13 TWIHS SleepWalking Matching Register

Name: TWIHS_SWMR
Offset: 0x4C
Reset: 0x00000000
Property: Read/Write

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

Bit	31	30	29	28	27	26	25	24
	DATAM[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	23	22	21	20	19	18	17	16
		SADR3[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
		SADR2[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		SADR1[6:0]						
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bits 31:24 – DATAM[7:0] Data Match
The TWIHS module extends the SleepWalking matching process to the first received data, comparing it with DATAM if DATAMEN bit is enabled.

Bits 22:16 – SADR3[6:0] Slave Address 3
Slave address 3. The TWIHS module matches on this additional address if SADR3EN bit is enabled.

Bits 14:8 – SADR2[6:0] Slave Address 2
Slave address 2. The TWIHS module matches on this additional address if SADR2EN bit is enabled.

Bits 6:0 – SADR1[6:0] Slave Address 1
Slave address 1. The TWIHS module matches on this additional address if SADR1EN bit is enabled.

Bit 5 – OVRUN Receive Overrun Interrupt Enable

Value	Description
0	No effect.
1	Enables the Receive Overrun Interrupt.

Bit 4 – RXRDY Receive Ready Interrupt Enable

Value	Description
0	No effect.
1	Enables the Receive Ready Interrupt.

Bit 1 – TXEMPTY Transmit Empty Interrupt Enable

Value	Description
0	No effect.
1	Enables the Transmit Empty Interrupt.

Bit 0 – TXRDY Transmit Ready Interrupt Enable

Value	Description
0	No effect.
1	Enables the Transmit Ready Interrupt.

46.7.41 USART LON Beta1 Rx Register

Name: US_LONB1RX
Offset: 0x0078
Reset: 0x0
Property: Read/Write

This register is relevant only if USART_MODE = 0x9 in the USART Mode Register.

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

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	BETA1RX[23:16]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	BETA1RX[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	BETA1RX[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bits 23:0 – BETA1RX[23:0] LON Beta1 Length after Reception

Value	Description
1–16777215	LON beta1 length after reception in t_{bit}.

48.7.9 HBI Channel Mask 0 Register

Name: MLB_HCMR0
Offset: 0x088
Reset: 0x00000000
Property: Read/Write

The HC can control which channel(s) are able to generate an HBI interrupt by writing the HBI Channel Mask Registers (HCMRn). Each bit of HCMRn is read/write.

Bit	31	30	29	28	27	26	25	24
	CHM: Bitwise Channel Mask Bit [31[31:24]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
	CHM: Bitwise Channel Mask Bit [31[23:16]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	CHM: Bitwise Channel Mask Bit [31[15:8]							
Access								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	CHM: Bitwise Channel Mask Bit [31[7:0]							
Access								
Reset	0	0	0	0	0	0	0	0

Bits 31:0 – CHM: Bitwise Channel Mask Bit [31[31:0] 0]
CHM[n] = 1 indicates that channel n can generate an interrupt.

50.7.1 TC Channel Control Register

Name: TC_CCRx
Offset: 0x00 + x*0x40 [x=0..2]
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
						SWTRG	CLKDIS	CLKEN
Access						W	W	W
Reset						–	–	–

Bit 2 – SWTRG Software Trigger Command

Value	Description
0	No effect.
1	A software trigger is performed: the counter is reset and the clock is started.

Bit 1 – CLKDIS Counter Clock Disable Command

Value	Description
0	No effect.
1	Disables the clock.

Bit 0 – CLKEN Counter Clock Enable Command

Value	Description
0	No effect.
1	Enables the clock if CLKDIS is not 1.

50.7.8 TC Register B

Name: TC_RBx
Offset: 0x18 + x*0x40 [x=0..2]
Reset: 0x00000000
Property: Read/Write

This register has access Read-only if TC_CMRx.WAVE = 0, Read/Write if TC_CMRx.WAVE = 1.
This register can only be written if the WPEN bit is cleared in the TC Write Protection Mode Register.

Bit	31	30	29	28	27	26	25	24
	RB[31:24]							
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
	RB[23:16]							
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
	RB[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
	RB[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 31:0 – RB[31:0] Register B
RB contains the Register B value in real time.

Important:
For 16-bit channels, RB field size is limited to register bits 15:0.

51.7.10 PWM DMA Register

Name: PWM_DMAR
Offset: 0x24
Reset: –
Property: Write-only

Only the first 16 bits (channel counter size) are significant.

Bit	31	30	29	28	27	26	25	24
Access								
Reset								

Bit	23	22	21	20	19	18	17	16
	DMADUTY[23:16]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0

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

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

Bits 23:0 – DMADUTY[23:0] Duty-Cycle Holding Register for DMA Access
Each write access to PWM_DMAR sequentially updates PWM_CDTYUPDx.CDTYUPD with DMADUTY (only for channel configured as synchronous). See “Method 3: Automatic write of duty-cycle values and automatic trigger of the update” .

52.7.13 AFEC Interrupt Status Register

Name: AFEC_ISR
Offset: 0x30
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
		TEMPCHG				COMPE	GOVRE	DRDY
Access		R				R	R	R
Reset		0				0	0	0

Bit	23	22	21	20	19	18	17	16
Access								
Reset								

Bit	15	14	13	12	11	10	9	8
					EOC11	EOC10	EOC9	EOC8
Access								
Reset					0	0	0	0

Bit	7	6	5	4	3	2	1	0
	EOC7	EOC6	EOC5	EOC4	EOC3	EOC2	EOC1	EOC0
Access								
Reset	0	0	0	0	0	0	0	0

Bit 30 – TEMPCHG Temperature Change (cleared on read)

Value	Description
0	No comparison match (defined in AFEC_TEMPCMPR) occurred since the last read of AFEC_ISR.
1	The temperature value reported on AFEC_CDR (AFEC_CSELR.CSEL = 11) has changed since the last read of AFEC_ISR, according to what is defined in the Temperature Mode register (AFEC_TEMPMR) and the Temperature Compare Window register (AFEC_TEMPCWR).

Bit 26 – COMPE Comparison Error (cleared by reading AFEC_ISR)

Value	Description
0	No comparison error since the last read of AFEC_ISR.
1	At least one comparison error has occurred since the last read of AFEC_ISR.

Bit 25 – GOVRE General Overrun Error (cleared by reading AFEC_ISR)

Value	Description
0	No general overrun error occurred since the last read of AFEC_ISR.
1	At least one general overrun error has occurred since the last read of AFEC_ISR.

58.12 Embedded Flash Characteristics

Table 58-50. Flash Characteristics

Parameter	Conditions		Min	Typ	Max	Unit
ERASE Line Assertion Time	–		230	–	–	ms
Program Cycle Time	Write Page Mode		–	1.5	–	ms
	Erase Page Mode		–	10	50	ms
	Erase Block Mode (by 8 Kbytes)		–	80	200	ms
	Erase Sector Mode			800	1500	ms
Full Chip Erase	512 Kbytes		–	3	6	s
	1 Mbytes		–	6	12	s
	2 Mbytes (only for SAMV71)		–	13	24	s
Data Retention	At T_A = 85°C, after 10K cycles (see **Note 1**)		10	–	–	Years
	At T_A = 85°C, after 1K cycles (see **Note 1**)		20	–	–	Years
	At T_A = 105°C, after 1K cycles (see **Note 1**)		5.5	–	–	Years
Endurance	Write/Erase cycles per page, block or sector at 25°C		100K			Cycles
	Write/Erase cycles per page, block or sector at 105°C		10K	–	–	Cycles
Flash Active Current	Random 128-bit read at maximum frequency at 25°C	on V_{DDCORE} =1.2V	–	16	20	mA
		on V_{DDIO}	–	2	10	
	Program at 25°C	on V_{DDCORE} =1.2V	–	2	3	
		on V_{DDIO}	–	8	12	
	Erase at 25°C	on V_{DDCORE} =1.2V	–	2	2	
		on V_{DDIO}	–	8	12	

Note:

1. Cycling over full temperature range.

Maximum operating frequencies are shown in the following table, but are limited by the Embedded Flash access time when the processor is fetching code out of it. These tables provide the device maximum operating frequency defined by the field FWS of the EEFC_FMR register. This field defines the number of wait states required to access the Embedded Flash Memory.

Table 58-51. Embedded Flash Wait States for Worst-Case Conditions

FWS	Read Operations	Maximum Operating Frequency (MHz) - VDDIO 3.0V
0	1 cycle	23
1	2 cycles	46
2	3 cycles	69
3	4 cycles	92
4	5 cycles	115
5	6 cycles	138
6	7 cycles	150

58.13 Timings for STH Conditions

58.13.1 AC Characteristics

58.13.1.1 Processor Clock Characteristics

Table 58-52. Processor Clock Waveform Parameters

Symbol	Parameter	Conditions	Min	Max	Unit
$1/(t_{CPPCK})$	Processor Clock Frequency	Worst case	–	300	MHz

58.13.1.2 Master Clock Characteristics

Table 58-53. Master Clock Waveform Parameters

Symbol	Parameter	Conditions	Min	Max	Unit
$1/(t_{CPMCK})$	Master Clock Frequency	Worst case	–	150	MHz

58.13.1.3 I/O Characteristics

Criteria used to define the maximum frequency of the I/Os:

- Output duty cycle (40%-60%)
- Minimum output swing: 100 mV to V_{DDIO} - 100 mV
- Addition of rising and falling time inferior to 75% of the period

Table 58-54. I/O Characteristics

Symbol	Parameter	Conditions			Min	Max	Unit
		Load	V_{DDIO}	Drive Level			
FreqMax1	Pin Group 1 (1) Maximum output frequency	10 pF	3.0V	Low	–	65	MHz
				High	–	115	
		25 pF		Low	–	28	