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	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	256K x 8
Voltage - Supply (Vcc/Vdd)	1.7V ~ 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/atsame70n19b-cn

17. SAM-BA Boot Program

17.1 Description

The SAM-BA Boot Program integrates an array of programs permitting download and/or upload into the different memories of the product.

17.2 Embedded Characteristics

- Default Boot Program
- Interface with SAM-BA Graphic User Interface
- SAM-BA Boot
  - Supports several communication media
    l Serial Communication on UART0

    l USB device port communication up to 1Mbyte/s
  - USB Requirements
    l External crystal or external clock with frequency of 12 MHz or 16 MHz

17.3 Hardware and Software Constraints

- SAM-BA Boot uses the first 2048 bytes of the SRAM for variables and stacks. The remaining available bytes can be used for user code.
- USB Requirements:
  - External crystal or external clock (see the following **Note**) with frequency of 12 MHz or 16 MHz

  Note: Must be 2500 ppm and VDDIO square wave signal.
- UART0 Requirements:
  - None. If no accurate external clock source is available, the internal 12 MHz RC meets RS-232 standards.

Table 17-1. Pins Driven during Boot Program Execution

Peripheral	Pin	PIO Line
UART0	URXD0	PA9
UART0	UTXD0	PA10

17.4 Flow Diagram

The boot program implements the algorithm below.

20.3 Register Summary

Offset	Name	Bit Pos.								
0x10	UTMI_OHCIICR	7:0			APPSTART	ARIE				RESx
		15:8								
		23:16	UDPPUDIS							
		31:24								
0x14 ... 0x2F	Reserved									
0x30	UTMI_CKTRIM	7:0							FREQ[1:0]	
		15:8								
		23:16								
		31:24								

- RTC Mode Register
- RTC Time Alarm Register
- RTC Calendar Alarm Register
- General Purpose Backup Registers
- Supply Controller Control Register
- Supply Controller Supply Monitor Mode Register
- Supply Controller Mode Register
- Supply Controller Wakeup Mode Register
- Supply Controller Wakeup Inputs Register

23.4.11 Register Bits in Backup Domain (VDDIO)

The following configuration registers, or certain bits of the registers, are physically located in the product backup domain:

- RSTC Mode Register (all bits)
- RTT Mode Register (all bits)
- RTT Alarm Register (all bits)
- RTC Control Register (all bits)
- RTC Mode Register (all bits)
- RTC Time Alarm Register (all bits)
- RTC Calendar Alarm Register (all bits)
- General Purpose Backup Registers (all bits)
- Supply Controller Control Register (see register description for details)
- Supply Controller Supply Monitor Mode Register (all bits)
- Supply Controller Mode Register (see register description for details)
- Supply Controller Wakeup Mode Register (all bits)
- Supply Controller Wakeup Inputs Register (all bits)
- Supply Controller Status Register (all bits)

28.5.1 Real-time Timer Mode Register

Name: RTT_MR
Offset: 0x00
Reset: 0x00008000
Property: Read/Write

Bit	31	30	29	28	27	26	25	24
								RTC1HZ
Access								
Reset								0

Bit	23	22	21	20	19	18	17	16
				RTTDIS		RTTRST	RTTINCIEN	ALMIEN
Access						R/W	R/W	R/W
Reset				0		0	0	0

Bit	15	14	13	12	11	10	9	8
	RTPRES[15:8]							
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	1	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
	RTPRES[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 24 – RTC1HZ Real-Time Clock 1Hz Clock Selection

Value	Description
0	The RTT 32-bit counter is driven by the 16-bit prescaler roll-over events.
1	The RTT 32-bit counter is driven by the 1Hz RTC clock.

Bit 20 – RTTDIS Real-time Timer Disable

Value	Description
0	The RTT is enabled.
1	The RTT is disabled (no dynamic power consumption).

Bit 18 – RTTRST Real-time Timer Restart

Value	Description
0	No effect.
1	Reloads and restarts the clock divider with the new programmed value. This also resets the 32-bit counter.

Bit 17 – RTTINCIEN Real-time Timer Increment Interrupt Enable

32.6.1.29 PIO Slow Clock Divider Debouncing Register

Name: PIO_SCDR
Offset: 0x008C
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
Access								
Reset								

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

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

Bits 13:0 – DIV[13:0] Slow Clock Divider Selection for Debouncing
$t_{div\_slck} = ((DIV + 1) \times 2) \times t_{slck}$

This field defines the delay between a Precharge Command and another Command in number of cycles. Number of cycles is between 0 and 15.

Bits 15:12 – TRC_TRFC[3:0] Row Cycle Delay and Row Refresh Cycle
Reset value is seven cycles.

This field defines two timings:

- the delay (t_{RFC}) between two Refresh commands and between a Refresh command and an Activate command
- the delay (t_{RC}) between two Active commands in number of cycles.

The number of cycles is between 0 and 15. The end user must program max {t_{RC}, t_{RFC}}.

Bits 11:8 – TWR[3:0] Write Recovery Delay
Reset value is two cycles.

This field defines the Write Recovery Time in number of cycles. Number of cycles is between 0 and 15.

Bit 7 – DBW Data Bus Width
Reset value is 16 bits.

This bit defines the Data Bus Width, which is 16 bits. It must be set to 1.

Value	Description
0	Data bus width is 32 bits.
1	Data bus width is 16 bits.

Bits 6:5 – CAS[1:0] CAS Latency
Reset value is two cycles. In the SDRAMC, only a CAS latency of two and three cycles is managed.

Value	Name	Description
0	Reserved	–
1	Reserved	–
1	LATENCY1	1 cycle latency
2	LATENCY2	2 cycle latency
3	LATENCY3	3 cycle latency

Bit 4 – NB Number of Banks
Reset value is two banks.

Value	Name	Description
0	BANK2	2 banks
1	BANK4	4 banks

Bits 3:2 – NR[1:0] Number of Row Bits
Reset value is 11 row bits.

Value	Name	Description
0	ROW11	11 bits to define the row number, up to 2048 rows
1	ROW12	12 bits to define the row number, up to 4096 rows
2	ROW13	13 bits to define the row number, up to 8192 rows
3	Reserved	

Offset	Name	Bit Pos.								
		23:16								
		31:24								
0x02B8	XDMAC_CC9	7:0	MEMSET	SWREQ		DSYNC		MBSIZE[1:0]		TYPE
		15:8		DIF	SIF	DWIDTH[1:0]		CSIZE[2:0]		
		23:16	WRIP	RDIP	INITD		DAM[1:0]		SAM[1:0]	
		31:24		PERID[6:0]						
0x02BC	XDMAC_CDS_MSP9	7:0	SDS_MSP[7:0]							
		15:8	SDS_MSP[15:8]							
		23:16	DDS_MSP[7:0]							
		31:24	DDS_MSP[15:8]							
0x02C0	XDMAC_CSUS9	7:0	SUBS[7:0]							
		15:8	SUBS[15:8]							
		23:16	SUBS[23:16]							
		31:24								
0x02C4	XDMAC_CDUS9	7:0	DUBS[7:0]							
		15:8	DUBS[15:8]							
		23:16	DUBS[23:16]							
		31:24								
0x02C8 ... 0x02CF	Reserved									
0x02D0	XDMAC_CIE10	7:0		ROIE	WBIE	RBIE	FIE	DIE	LIE	BIE
		15:8								
		23:16								
		31:24								
0x02D4	XDMAC_CID10	7:0		ROID	WBEID	RBEID	FID	DID	LID	BID
		15:8								
		23:16								
		31:24								
0x02D8	XDMAC_CIM10	7:0		ROIM	WBEIM	RBEIM	FIM	DIM	LIM	BIM
		15:8								
		23:16								
		31:24								
0x02DC	XDMAC_CIS10	7:0		ROIS	WBEIS	RBEIS	FIS	DIS	LIS	BIS
		15:8								
		23:16								
		31:24								
0x02E0	XDMAC_CSA10	7:0	SA[7:0]							
		15:8	SA[15:8]							
		23:16	SA[23:16]							
		31:24	SA[31:24]							
0x02E4	XDMAC_CDA10	7:0	DA[7:0]							
		15:8	DA[15:8]							
		23:16	DA[23:16]							
		31:24	DA[31:24]							
0x02E8	XDMAC_CNDA10	7:0	NDA[5:0]							NDAIF

37.6.25 ISI Write Protection Status Register

Name: ISI_WPSR
Offset: 0xE8
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
	WPVSRC[15:8]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	WPVSRC[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	0	0

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

Bits 23:8 – WPVSRC[15:0] Write Protection Violation Source

Value	Name
0	No Write Protection Violation occurred since the last read of this register (ISI_WPSR).
1	Write access in ISI_CFG1 while Write Protection was enabled (since the last read).
2	Write access in ISI_CFG2 while Write Protection was enabled (since the last read).
3	Write access in ISI_PSIZE while Write Protection was enabled (since the last read).
4	Write access in ISI_PDECF while Write Protection was enabled (since the last read).
5	Write access in ISI_Y2R_SET0 while Write Protection was enabled (since the last read).
6	Write access in ISI_Y2R_SET1 while Write Protection was enabled (since the last read).
7	Write access in ISI_R2Y_SET0 while Write Protection was enabled (since the last read).
8	Write access in ISI_R2Y_SET1 while Write Protection was enabled (since the last read).
9	Write access in ISI_R2Y_SET2 while Write Protection was enabled (since the last read).

Bit 0 – WPVS Write Protection Violation Status

Value	Name
0	No write protection violation occurred since the last read of ISI_WPSR.
1	A write protection violation has occurred since the last read of the ISI_WPSR. If this violation is an unauthorized attempt to write a protected register, the associated violation is reported into field WPVSRC.

To accommodate the status and statistics associated with each frame, three words per packet (or two if configured in 64-bit datapath mode) are reserved at the end of the packet data. If the packet is bad and requires to be dropped, the status and statistics are the only information held on that packet.

The receiver packet buffer will also detect a full condition so that an overflow condition can be detected. If this occurs, subsequent packets are dropped and an RX overflow interrupt is raised.

For full store and forward, the DMA only begins packet fetches once the status and statistics for a frame are available. If the frame has a bad status due to a frame error, the status and statistics are passed on to the GMAC registers. If the frame has a good status, the information is used to read the frame from the packet buffer memory and burst onto the AHB using the DMA buffer management protocol. Once the last frame data has been transferred to the packet buffer, the status and statistics are updated to the GMAC registers.

If Partial Store and Forward mode is active, the DMA will begin fetching the packet data before the status is available. As soon as the status becomes available, the DMA will fetch this information as soon as possible before continuing to fetch the remainder of the frame. Once the last frame data has been transferred to the packet buffer, the status and statistics are updated to the GMAC registers.

38.6.3.9 Priority Queuing in the DMA

The DMA by default uses a single transmit and receive queue. This means the list of transmit/receive buffer descriptors point to data buffers associated with a single transmit/receive data stream. The GMAC can select up to 6 priority queues. Each queue has an independent list of buffer descriptors pointing to separate data streams.

The table below gives the DPRAM size associated with each queue.

Table 38-4. Queue Size

Queue Number	Queue Size
5 (highest priority)	1 KB
4	2 KB
3	2 KB
2	512 bytes
1	512 bytes
0 (lowest priority)	2 KB

In the transmit direction, higher priority queues are always serviced before lower priority queues, with Q0 as lowest priority and Q5 as highest priority. This strict priority scheme requires the user to ensure that high priority traffic is constrained so that lower priority traffic will have required bandwidth. The GMAC DMA will determine the next queue to service by initiating a sequence of buffer descriptor reads interrogating the ownership bits of each. The buffer descriptor corresponding to the highest priority queue is read first.

As an example, if the ownership bit of this descriptor is set, the DMA will progress by reading the 2nd highest priority queue’s descriptor. If that ownership bit read of this lower priority queue is set as well, the DMA will read the 3rd highest priority queue’s descriptor. If all the descriptors return an ownership bit set, a resource error has occurred, so an interrupt is generated and transmission is automatically halted. Transmission can only be restarted by writing a '1' to the Transmission Start bit in the Network Control register (GMAC_NCR.TSTART). The GMAC DMA will need to identify the highest available queue to

39.6.4 General Status Set Register

Name: USBHS_SFR
Offset: 0x080C
Property: Write-only

This register always reads as zero.

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
							VBUSRQS	
Access								
Reset								

Bit	7	6	5	4	3	2	1	0
				RDERRIS				
Access								
Reset								

Bit 9 – VBUSRQS VBUS Request Set
Must be set to ‘1’.

Value	Description
0	No effect.
1	Sets the VBUSRQ bit in USBHS_SR.

Bit 4 – RDERRIS Remote Device Connection Error Interrupt Set

Value	Description
0	No effect.
1	Sets the RDERRI bit in USBHS_SR, which may be useful for test or debug purposes.

Value	Description
0	The UTMI transceiver is in Normal operating mode.
1	The UTMI transceiver generates high-speed J state for test purposes.

Bit 12 – LS Low-Speed Mode Force
This bit can be written even if USBHS_CTRL.USBE = 0 or USBHS_CTRL.FRZCLK = 1. Disabling the USBHS (by writing a zero to the USBHS_CTRL.USBE bit) does not reset this bit.

Value	Description
0	The Full-speed mode is active.
1	The Low-speed mode is active.

Bits 11:10 – SPDCONF[1:0] Mode Configuration
This field contains the peripheral speed:

Value	Name	Description
0	NORMAL	The peripheral starts in Full-speed mode and performs a high-speed reset to switch to High-speed mode if the host is high-speed-capable.
1	LOW_POWER	For a better consumption, if high speed is not needed.
2	HIGH_SPEED	Forced high speed.
3	FORCED_FS	The peripheral remains in Full-speed mode whatever the host speed capability.

Bit 9 – RMWKUP Remote Wakeup
This bit is cleared when the USBHS receives a USB reset or once the upstream resume has been sent.

Value	Description
0	No effect.
1	Sends an upstream resume to the host for a remote wakeup.

Bit 8 – DETACH Detach

Value	Description
0	Reconnects the device.
1	Physically detaches the device (disconnects the internal pull-up resistor from D+ and D-).

Bit 7 – ADDEN Address Enable
This bit is cleared when a USB reset is received.

Value	Description
0	No effect.
1	Activates the UADD field (USB address).

Bits 6:0 – UADD[6:0] USB Address
This field contains the device address.

This field is cleared when a USB reset is received.

41.8 Register Summary

Offset	Name	Bit Pos.								
0x00	SPI_CR	7:0	SWRST						SPIDIS	SPIEN
		15:8				REQCLR				
		23:16								
		31:24								LASTXFER
0x04	SPI_MR	7:0	LLB		WDRBT	MODFDIS		PCSDEC	PS	MSTR
		15:8								
		23:16					PCS[3:0]			
		31:24	DLYBCS[7:0]							
0x08	SPI_RDR	7:0	RD[7:0]							
		15:8	RD[15:8]							
		23:16					PCS[3:0]			
		31:24								
0x0C	SPI_TDR	7:0	TD[7:0]							
		15:8	TD[15:8]							
		23:16					PCS[3:0]			
		31:24								LASTXFER
0x10	SPI_SR	7:0					OVRES	MODF	TDRE	RDRF
		15:8				SFERR		UNDES	TXEMPTY	NSSR
		23:16								SPIENS
		31:24								
0x14	SPI_IER	7:0					OVRES	MODF	TDRE	RDRF
		15:8						UNDES	TXEMPTY	NSSR
		23:16								
		31:24								
0x18	SPI_IDR	7:0					OVRES	MODF	TDRE	RDRF
		15:8						UNDES	TXEMPTY	NSSR
		23:16								
		31:24								
0x1C	SPI_IMR	7:0					OVRES	MODF	TDRE	RDRF
		15:8						UNDES	TXEMPTY	NSSR
		23:16								
		31:24								
0x20 ... 0x2F	Reserved									
0x30	SPI_CSR0	7:0	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL
		15:8	SCBR[7:0]							
		23:16	DLYBS[7:0]							
		31:24	DLYBCT[7:0]							
0x34	SPI_CSR1	7:0	BITS[3:0]				CSAAT	CSNAAT	NCPHA	CPOL
		15:8	SCBR[7:0]							
		23:16	DLYBS[7:0]							
		31:24	DLYBCT[7:0]							

Value	Name	Description
2	TRSFR_WRITE	Write transfer into the serial memory. Scrambling is not performed.
3	TRSFR_WRITE_MEMORY	Write data transfer into the serial memory. If enabled, scrambling is performed.

Bit 10 – ADDRL Address Length
The ADDRL bit determines the length of the address.

0 (24_BIT): The address is 24 bits long.

1 (32_BIT): The address is 32 bits long.

Bits 9:8 – OPTL[1:0] Option Code Length
The OPTL field determines the length of the option code. The value written in OPTL must be consistent with the value written in the field WIDTH. For example, OPTL = 0 (1-bit option code) is not consistent with WIDTH = 6 (option code sent with QuadSPI protocol, thus the minimum length of the option code is 4 bits).

Value	Name	Description
0	OPTION_1BIT	The option code is 1 bit long.
1	OPTION_2BIT	The option code is 2 bits long.
2	OPTION_4BIT	The option code is 4 bits long.
3	OPTION_8BIT	The option code is 8 bits long.

Bit 7 – DATAEN Data Enable

Value	Description
0	No data is sent/received to/from the serial Flash memory.
1	Data is sent/received to/from the serial Flash memory.

Bit 6 – OPTEN Option Enable

Value	Description
0	The option is not sent to the serial Flash memory.
1	The option is sent to the serial Flash memory.

Bit 5 – ADDREN Address Enable

Value	Description
0	The transfer address is not sent to the serial Flash memory.
1	The transfer address is sent to the serial Flash memory.

Bit 4 – INSTEN Instruction Enable

Value	Description
0	The instruction is not sent to the serial Flash memory.
1	The instruction is sent to the serial Flash memory.

Bits 2:0 – WIDTH[2:0] Width of Instruction Code, Address, Option Code and Data

43.7.8 TWIHS Filter Register

Name: TWIHS_FILTR
Offset: 0x44
Reset: 0x00000000
Property: Read/Write

TWIHS digital input filtering follows a majority decision based on three samples from SDA/SCL lines at peripheral clock frequency.

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						THRES[2:0]		
Reset						0	0	0

Bit	7	6	5	4	3	2	1	0
Access						PADFCFG	PADFEN	FILT
Reset						0	0	0

Bits 10:8 – THRES[2:0] Digital Filter Threshold

Value	Description
0	No filtering applied on TWIHS inputs.
1-7	Maximum pulse width of spikes to be suppressed by the input filter, defined in peripheral clock cycles.

Bit 2 – PADFCFG PAD Filter Config
See the electrical characteristics section for filter configuration details.

Bit 1 – PADFEN PAD Filter Enable

Value	Description
0	PAD analog filter is disabled.
1	PAD analog filter is enabled. (The analog filter must be enabled if High-speed mode is enabled.)

Bit 0 – FILT RX Digital Filter

46.6.10.5.8 End Of Frame

The USART configured in LON mode terminates the frame with a 3 t_{bit} long Manchester code violation. After sending the last CRC bit it maintains the data transitionless during three bit periods.

46.6.10.6 LON Operating Modes

46.6.10.6.1 Transmitting/Receiving Modules

According to the LON node configuration and LON network state, the transmitting module will be activated if a transmission request has been made and access to the LON bus granted. It returns to idle state once the transmission ends.

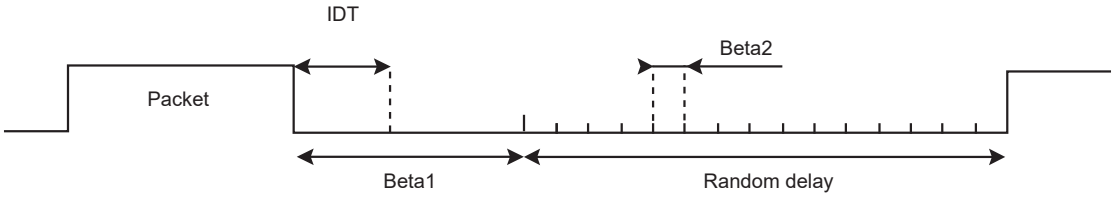
According to the LON node configuration and LON network state, the receiving module will be activated if a valid preamble is detected and the transmitting module is not activated.

46.6.10.6.2 comm_type

In the CEA-709 standard, two communication configurations are defined and configurable through the comm_type variable. The comm_type variable value can be set in the USART LON Mode register (US_LONMR) through the COMMT bit. The selection of the comm_type determines the MAC behavior in the following ways:

- comm_type=1:
  - An indeterminate time is defined during the Beta 1 period in which all transitions on the channel are ignored, as shown in Figure 46-58.
  - The MAC sublayer ignores collisions occurring during the first 25% of the transmitted preamble. It optionally (according to US_LONMR.CDTAIL) ignores collisions reported following the transmission of the CRC but prior to the end of transmission.
  - If a collision is detected during preamble transmission, the MAC sublayer can terminate the packet if so configured according to US_LONMR.TCOL. Collisions detected after the preamble has been sent do not terminate transmission.
- comm_type=2:
  - No indeterminate time is defined at the MAC sublayer.
  - The MAC sublayer shall always terminate the packet upon notification of a collision.

Figure 46-58. LON Indeterminate Time



46.6.10.6.3 Collision Detection

As an option of the CEA-709 standard, collision detection is supported through an active low Collision Detect (CD) input from the transceiver.

The Collision Detection source can be either external (See "I/O Lines Description") or internal. The collision detection source selection is defined through US_LONMR.LCDS.

The Collision Detection feature can be activated through US_LONMR.COLDET. If the collision detection feature is enabled and CD signal goes low for at least half t_{bit} period then a collision is detected and reported as defined in "comm_type".

46.6.10.6.4 Collision Detection Mode.

As defined in "comm_type", if comm_type=1 the LON node can be either configured to not terminate transmission upon collision notification during preamble transmission or terminate transmission.

US_LONMR.TCOL determines whether to terminate transmission or not upon collision notification during preamble transmission.

46.6.10.6.5 Collision Detection After CRC

As defined in “comm_type” on page 64, if comm_type=1 the LON node can be either be configured to ignore collision after the CRC has been sent but prior to the end of the frame.

US_LONMR.CDTAIL determines whether such collision notifications must be considered or not.

46.6.10.6.6 Random Number Generation

The Predictive p-persistent CSMA algorithm defined in the CEA-709.1 Standard is based on a random number generation.

This random number is automatically generated by an internal algorithm.

In addition, a USART IC DIFF register (US_ICDIFF) is available to avoid that two same chips with the same software generate the same random number after reset. The value of this register is used by the internal algorithm to generate the random number. Therefore, putting a different value here for each chip ensures that the random number generated after a reset at the same time, will not be the same. It is recommended to put the chip ID code here.

46.6.10.7 LON Node Backlog Estimation

As defined in the CEA-709 standard, the LON node maintains its own backlog estimation. The node backlog estimation is initially set to 1, will always be greater than 1 and will never exceed 63. If the node backlog estimation exceeds the maximum backlog value, the backlog value is set to 63 and a backlog overflow error flag is set (LBLOVFE flag).

The node backlog estimation is incremented each time a frame is sent or received successfully. The increment to the backlog is encoded into the link layer header, and represents the number of messages that the packet shall cause to be generated upon reception.

The backlog decrements under one of the following conditions:

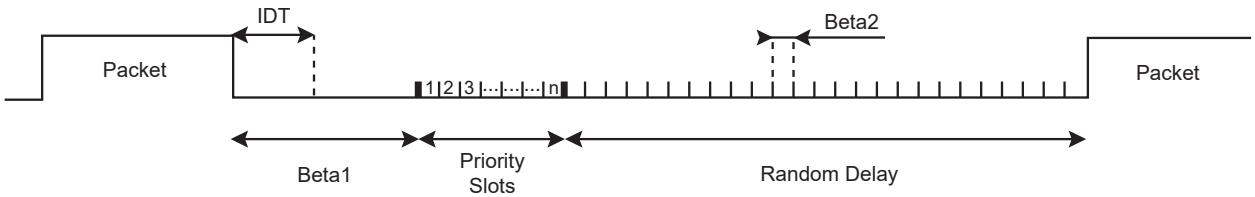
- On waiting to transmit: If Wbase randomizing slots go by without channel activity.
- On receive: If a packet is received with a backlog increment of ‘0’.
- On transmit: If a packet is transmitted with a backlog increment of ‘0’.
- On idle: If a packet cycle time expires without channel activity.

46.6.10.7.1 Optional Collision Detection Feature And Backlog Estimation

Each time a frame is transmitted and a collision occurred, the backlog is incremented by 1. In this case, the backlog increment encoded in the link layer is ignored.

46.6.10.8 LON Timings

Figure 46-59. LON Timings



46.6.10.8.1 Beta2

A node wishing to transmit generates a random delay T. This delay is an integer number of randomizing slots of duration Beta2.

The beta2 length (in t_{bit}) is configurable through US_FIDI. Note that a length of ‘0’ is not allowed.

47.6.6 UART Status Register

Name: UART_SR
Offset: 0x14
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
	CMP						TXEMPTY	
Access	R						R	
Reset	0						0	

Bit	7	6	5	4	3	2	1	0
	PARE	FRAME	OVRE				TXRDY	RXRDY
Access	R	R	R				R	R
Reset	0	0	0				0	0

Bit 15 – CMP Comparison Match

Value	Description
0	No received character matches the comparison criteria programmed in VAL1, VAL2 fields and in CMPPAR bit since the last RSTSTA.
1	The received character matches the comparison criteria.

Bit 9 – TXEMPTY Transmitter Empty

Value	Description
0	There are characters in UART_THR, or characters being processed by the transmitter, or the transmitter is disabled.
1	There are no characters in UART_THR and there are no characters being processed by the transmitter.

Bit 7 – PARE Parity Error

Value	Description
0	No parity error has occurred since the last RSTSTA.
1	At least one parity error has occurred since the last RSTSTA.

Bit 6 – FRAME Framing Error

1: Dual ID filter for EF1ID or EF2ID

2: Classic filter: EF1ID = filter, EF2ID = mask

3: Range filter from EF1ID to EF2ID (EF2ID ≥ EF1ID), MCAN_XIDAM mask not applied

• F1 Bits 28:0 EFID2[28:0]: Extended Filter ID 2

This field has a different meaning depending on the configuration of EFEC:

• EFEC = “001”...“110”–Second ID of extended ID filter element

• EFEC = “111”–Filter for Rx Buffers or for debug messages

EFID2[10:9] decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence.

0: Store message in an Rx buffer

1: Debug Message A

2: Debug Message B

3: Debug Message C

EFID2[5:0] defines the index of the dedicated Rx Buffer element to which a matching message is stored.

49.5.8 Hardware Reset Description

After hardware reset, the registers of the MCAN hold the reset values listed in the register descriptions. Additionally the Bus_Off state is reset and the output CANTX is set to recessive (HIGH). The value 0x0001 (MCAN_CCCR.INIT = ‘1’) in the CC Control register enables software initialization. The MCAN does not influence the CAN bus until the processor resets MCAN_CCCR.INIT to ‘0’.

49.5.9 Access to Reserved Register Addresses

In case the application software accesses one of the reserved addresses in the MCAN register map (read or write access), interrupt flag MCAN_IR.ARA is set and, if enabled, the selected interrupt line is risen.

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.

52.6.12 AFE Timings

Each AFE has its own minimal startup time configured in AFEC_MR.STARTUP.

WARNING No input buffer amplifier to isolate the source is included in the AFE. This must be taken into consideration.

52.6.13 Temperature Sensor

The temperature sensor is internally connected to channel index 11.

The AFEC manages temperature measurement in several ways. The different methods of measurement depend on the configuration bits TRGEN in the AFEC_MR and CH11 in AFEC_CHSR.

Temperature measurement can be triggered at the same rate as other channels by enabling the conversion channel 11.

If AFEC_CHSR.CH11 is enabled, the temperature sensor analog cell is switched on. If a user sequence is used, the last converted channel of the sequence is always the temperature sensor channel.

A manual start can be performed only if AFEC_MR.TRGEN is disabled. When AFEC_CR.START is set, the temperature sensor channel conversion is scheduled together with the other enabled channels (if any). The result of the conversion is placed in an internal register that can be read in the AFEC_CDR (AFEC_CSELR must be programmed accordingly prior to reading AFEC_CDR) and the associated flag EOC11 is set in the AFEC_ISR.

The channel of the temperature sensor is periodically converted together with the other enabled channels and the result is placed into AFEC_LCDR and an internal register (can be read in AFEC_CDR). Thus the temperature conversion result is part of the Peripheral DMA Controller buffer. The temperature channel can be enabled/disabled at any time, but this may not be optimal for downstream processing.