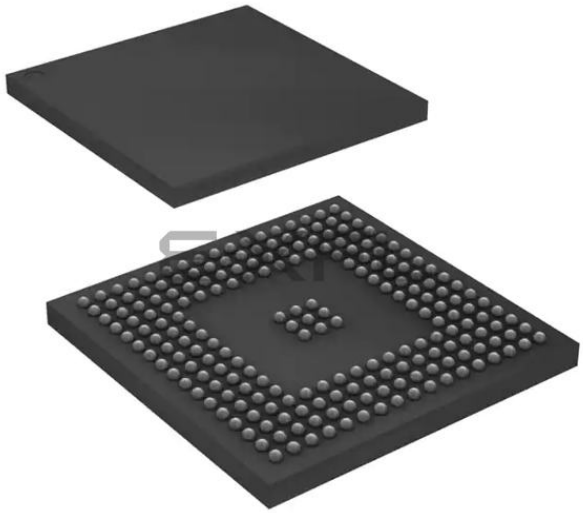


Welcome to E-XFL.COM

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	Active
Core Processor	ARM926EJ-S
Number of Cores/Bus Width	1 Core, 32-Bit
Speed	400MHz
Co-Processors/DSP	-
RAM Controllers	DDR2, SDRAM, SRAM
Graphics Acceleration	No
Display & Interface Controllers	-
Ethernet	10/100Mbps
SATA	-
USB	USB 2.0 (3)
Voltage - I/O	1.8V, 3.3V
Operating Temperature	-40°C ~ 85°C (TA)
Security Features	-
Package / Case	217-LFBGA
Supplier Device Package	217-LFBGA (15x15)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at91sam9g25-cu

Table 2-1. Signal Description List (Continued)

Signal Name	Function	Type	Active Level
USB Host High Speed Port - UHPHS			
HFSDPA	USB Host Port A Full Speed Data +	Analog	
HFSDMA	USB Host Port A Full Speed Data -	Analog	
HHSDPA	USB Host Port A High Speed Data +	Analog	
HHSDMA	USB Host Port A High Speed Data -	Analog	
HFSDPB	USB Host Port B Full Speed Data +	Analog	
HFSDMB	USB Host Port B Full Speed Data -	Analog	
HHSDPB	USB Host Port B High Speed Data +	Analog	
HHSDMB	USB Host Port B High Speed Data -	Analog	
HFSDMC	USB Host Port C Full Speed Data -	Analog	
HFSDPC	USB Host Port C Full Speed Data +	Analog	
Ethernet 10/100 - EMAC			
ETXCK	Transmit Clock or Reference Clock	Input	
ERXCK	Receive Clock	Input	
ETXEN	Transmit Enable	Output	
ETX0–ETX3	Transmit Data	Output	
ETXER	Transmit Coding Error	Output	
ERXDV	Receive Data Valid	Input	
ERX0–ERX3	Receive Data	Input	
ERXER	Receive Error	Input	
ECRS	Carrier Sense and Data Valid	Input	
ECOL	Collision Detect	Input	
EMDC	Management Data Clock	Output	
EMDIO	Management Data Input/Output	I/O	
Analog-to-Digital Converter - ADC			
AD0–AD11	12 Analog Inputs	Analog	
ADTRG	ADC Trigger	Input	
ADVREF	ADC Reference	Analog	
Soft Modem Device - SMD			
DIBN	Soft Modem Signal	I/O	
DIBP	Soft Modem Signal	I/O	

12.9.9 AIC Interrupt Enable Command Register

Name: AIC_IECR

Address: 0xFFFFF120

Access: Write-only

31	30	29	28	27	26	25	24
PID31	PID30	PID29	PID28	PID27	PID26	PID25	PID24
23	22	21	20	19	18	17	16
PID23	PID22	PID21	PID20	PID19	PID18	PID17	PID16
15	14	13	12	11	10	9	8
PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8
7	6	5	4	3	2	1	0
PID7	PID6	PID5	PID4	PID3	PID2	SYS	FIQ

- **FIQ: Interrupt Enable**

0: No effect.

1: Enables corresponding interrupt.

- **SYS: Interrupt Enable**

0: No effect.

1: Enables corresponding interrupt.

- **PID2–PID31: Interrupt Enable**

0: No effect.

1: Enables corresponding interrupt.

12.9.18 AIC Fast Forcing Status Register

Name: AIC_FFSR

Address: 0xFFFFF148

Access: Read-only

31	30	29	28	27	26	25	24
PID31	PID30	PID29	PID28	PID27	PID26	PID25	PID24
23	22	21	20	19	18	17	16
PID23	PID22	PID21	PID20	PID19	PID18	PID17	PID16
15	14	13	12	11	10	9	8
PID15	PID14	PID13	PID12	PID11	PID10	PID9	PID8
7	6	5	4	3	2	1	0
PID7	PID6	PID5	PID4	PID3	PID2	SYS	–

- **SYS: Fast Forcing Status**

0: The Fast Forcing feature is disabled on the corresponding interrupt.

1: The Fast Forcing feature is enabled on the corresponding interrupt.

- **PID2–PID31: Fast Forcing Status**

0: The Fast Forcing feature is disabled on the corresponding interrupt.

1: The Fast Forcing feature is enabled on the corresponding interrupt.

13. Reset Controller (RSTC)

13.1 Description

The Reset Controller (RSTC), based on power-on reset cells, handles all the resets of the system without any external components. It reports which reset occurred last.

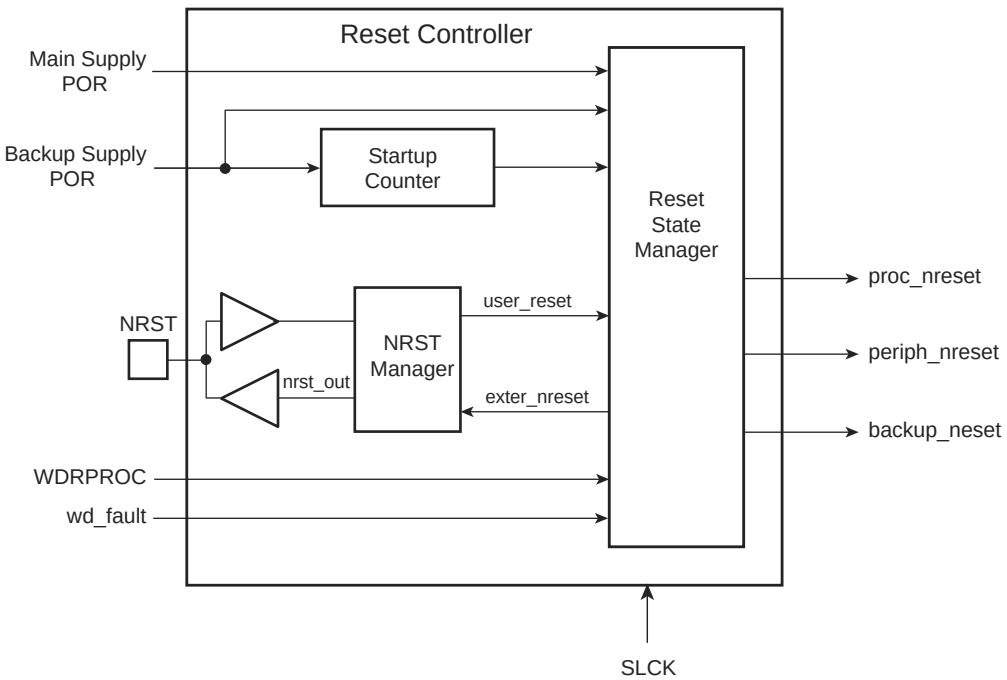
The Reset Controller also drives independently or simultaneously the external reset and the peripheral and processor resets.

13.2 Embedded Characteristics

- Manages All Resets of the System, Including
  - External Devices Through the NRST Pin
  - Processor Reset
  - Peripheral Set Reset
  - Backed-up Peripheral Reset
- Based on 2 Embedded Power-on Reset Cells
- Reset Source Status
  - Status of the Last Reset
  - Either General Reset, Wake-up Reset, Software Reset, User Reset, Watchdog Reset
- External Reset Signal Shaping

13.3 Block Diagram

Figure 13-1. Reset Controller Block Diagram



14.6.4 RTC Calendar Register

Name: RTC_CALR

Address: 0xFFFFFEBC

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	DATE					

23	22	21	20	19	18	17	16
DAY			MONTH				

15	14	13	12	11	10	9	8
YEAR							

7	6	5	4	3	2	1	0
–	CENT						

- **CENT: Current Century**

Only the BCD value 20 can be configured.

The lowest four bits encode the units. The higher bits encode the tens.

- **YEAR: Current Year**

The range that can be set is 00–99 (BCD).

The lowest four bits encode the units. The higher bits encode the tens.

- **MONTH: Current Month**

The range that can be set is 01–12 (BCD).

The lowest four bits encode the units. The higher bits encode the tens.

- **DAY: Current Day in Current Week**

The range that can be set is 1–7 (BCD).

The coding of the number (which number represents which day) is user-defined as it has no effect on the date counter.

- **DATE: Current Day in Current Month**

The range that can be set is 01–31 (BCD).

The lowest four bits encode the units. The higher bits encode the tens.

18.3.1 General Purpose Backup Register x

Name: SYS_GPBRx

Address: 0xFFFFFE60

Access: Read/Write

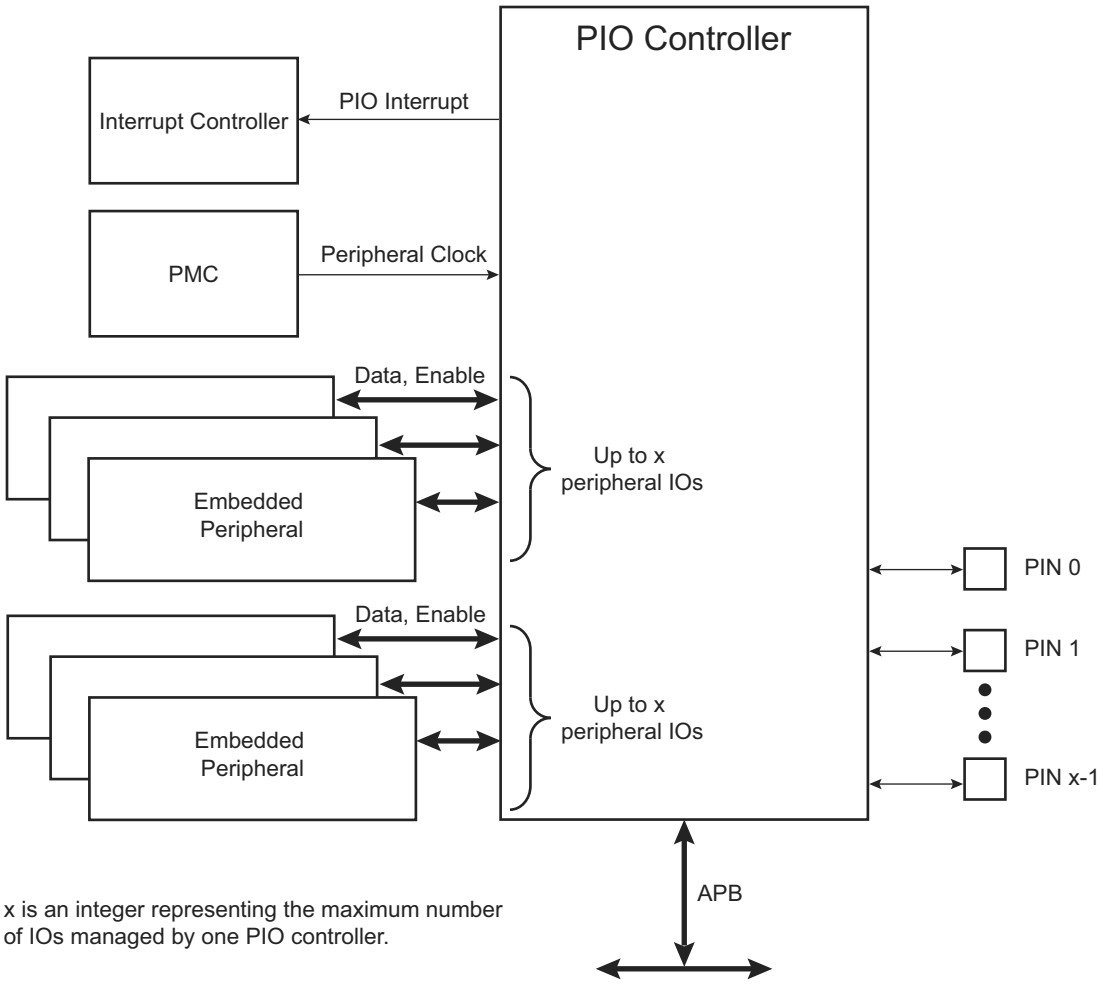
31	30	29	28	27	26	25	24
GPBR_VALUE							
23	22	21	20	19	18	17	16
GPBR_VALUE							
15	14	13	12	11	10	9	8
GPBR_VALUE							
7	6	5	4	3	2	1	0
GPBR_VALUE							

These registers are reset at first power-up and on each loss of VDDBU.

- **GPBR_VALUE: Value of GPBR x**

22.3 Block Diagram

Figure 22-1. Block Diagram



22.4 Product Dependencies

22.4.1 Pin Multiplexing

Each pin is configurable, depending on the product, as either a general-purpose I/O line only, or as an I/O line multiplexed with one or two peripheral I/Os. As the multiplexing is hardware defined and thus product-dependent, the hardware designer and programmer must carefully determine the configuration of the PIO Controllers required by their application. When an I/O line is general-purpose only, i.e., not multiplexed with any peripheral I/O, programming of the PIO Controller regarding the assignment to a peripheral has no effect and only the PIO Controller can control how the pin is driven by the product.

22.4.2 External Interrupt Lines

The interrupt signals FIQ and IRQ0 to IRQn are generally multiplexed through the PIO Controllers. However, it is not necessary to assign the I/O line to the interrupt function as the PIO Controller has no effect on inputs and the external interrupt lines are used only as inputs.

When the WKUPx input pins must be used as external interrupt lines, the PIO Controller must be configured to disable the peripheral control on these IOs, and the corresponding IO lines must be set to Input mode.

22.6.37 PIO Additional Interrupt Modes Disable Register

Name: PIO_AIMDR

Address: 0xFFFFF4B4 (PIOA), 0xFFFFF6B4 (PIOB), 0xFFFFF8B4 (PIOC), 0xFFFFFAB4 (PIOD)

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: Additional Interrupt Modes Disable**

0: No effect.

1: The interrupt mode is set to the default interrupt mode (both-edge detection).

```
/* In either case compute delta */
delta[i+1]  = (mu[i+1] * 2 - lmu[i+1]) >> 1;
/* In either case compute the discrepancy */
for (k = 0 ; k <= (lmu[i+1]>>1); k++)
{
      if (k == 0)
      dmu[i+1] = si[2*(i-1)+3];
      /* check if one operand of the multiplier is null, its index is -1 */
      else if (smu[i+1][k] && si[2*(i-1)+3-k])
      dmu[i+1] = gf_antilog[(gf_log[smu[i+1][k]] + gf_log[si[2*(i-1)+3-
k]])%nn] ^ dmu[i+1];
}
}
return 0;
}
```

26.5.3 Find the Error Position

The output of the get_sigma() procedure is a polynomial stored in the smu[NB_ERROR+1][] table. The error position is the roots of that polynomial. The degree of this polynomial is very important information, as it gives the number of errors. The PMERRLOC module provides a hardware accelerator for this step.

28.14 Asynchronous Page Mode

The SMC supports asynchronous burst reads in page mode, providing that the page mode is enabled in the SMC_MODE register (PMEN field). The page size must be configured in the SMC_MODE register (PS field) to 4, 8, 16 or 32 bytes.

The page defines a set of consecutive bytes into memory. A 4-byte page (resp. 8-, 16-, 32-byte page) is always aligned to 4-byte boundaries (resp. 8-, 16-, 32-byte boundaries) of memory. The MSB of data address defines the address of the page in memory, the LSB of address define the address of the data in the page as detailed in Table 28-7.

With page mode memory devices, the first access to one page (t_{pa}) takes longer than the subsequent accesses to the page (t_{sa}) as shown in Figure 28-34. When in page mode, the SMC enables the user to define different read timings for the first access within one page, and next accesses within the page.

Table 28-7. Page Address and Data Address within a Page

Page Size	Page Address (1)	Data Address in the Page (2)
4 bytes	A[25:2]	A[1:0]
8 bytes	A[25:3]	A[2:0]
16 bytes	A[25:4]	A[3:0]
32 bytes	A[25:5]	A[4:0]

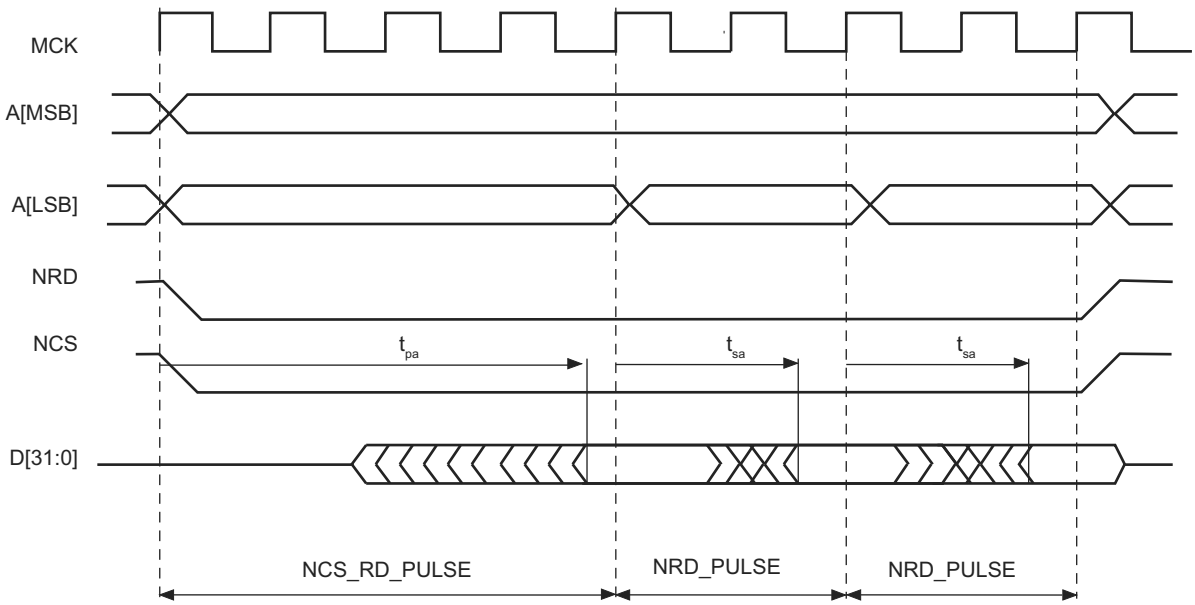
Notes: 1. 'A' denotes the address bus of the memory device

2. For 16-bit devices, the bit 0 of address is ignored. For 32-bit devices, bits [1:0] are ignored.

28.14.1 Protocol and Timings in Page Mode

Figure 28-34 shows the NRD and NCS timings in page mode access.

Figure 28-34. Page Mode Read Protocol (Address MSB and LSB are defined in Table 28-7)



The NRD and NCS signals are held low during all read transfers, whatever the programmed values of the setup and hold timings in the User Interface may be. Moreover, the NRD and NCS timings are identical. The pulse length of the first access to the page is defined with the NCS_RD_PULSE field of the SMC_PULSE register. The pulse length of subsequent accesses within the page are defined using the NRD_PULSE parameter.

a. Read the channel register to choose an available (disabled) channel.
b. Clear any pending interrupts on the channel from the previous DMA transfer by reading the DMAC_EBCISR.
c. Program the channel registers in the Memory with the first descriptor. This descriptor will be word oriented. This descriptor is referred to as LLI_W(n), standing for LLI word oriented transfer for block *n*.
d. The LLI_W(n).DMAC_SADDRx field in memory must be set with the starting address of the HSMCI_FIFO address.
e. The LLI_W(n).DMAC_DADDRx field in the memory must be word aligned.
f. Configure the fields of LLI_W(n).DMAC_CTRLAx as follows:
–DST_WIDTH is set to WORD
–SRC_WIDTH is set to WORD
–SCSIZE must be set according to the value of HSMCI_DMA.CHKSIZE.
–BTSIZE is programmed with *block_length/4.*
g. Configure the fields of LLI_W(n).DMAC_CTRLBx as follows:
–DST_INCR is set to INCR.
–SRC_INCR is set to INCR.
–FC field is programmed with peripheral to memory flow control mode.
–SRC_DSCR is configured to 0 (descriptor fetch is enabled for the SRC).
–DST_DSCR is set to TRUE (descriptor fetch is disabled for the DST).
–DIF and SIF are set with their respective layer ID. If SIF is different from DIF, the DMA Controller is able to prefetch data and write HSMCI simultaneously.
h. Configure the fields of the LLI_W(n).DMAC_CFGx register for Channel x as follows:
–FIFOCFG defines the watermark of the DMA channel FIFO.
–DST_REP is set to zero. Addresses are contiguous.
–SRC_H2SEL is set to true to enable hardware handshaking on the destination.
–SRC_PER is programmed with the hardware handshaking ID of the targeted HSMCI Host Controller.
i. Program LLI_W(n).DMAC_DSCRx with the address of LLI_W(n+1) descriptor. And set the DSCRx_IF to the AHB Layer ID. This operation actually links descriptors together. If LLI_W(n) is the last descriptor then LLI_W(n).DMAC_DSCRx points to 0.
j. Program the DMAC_CTRLBx register for Channel x with 0. Its content is updated with the LLI Fetch operation.
k. Program DMAC_DSCRx for Channel x with the address of LLI_W(0).
l. Enable Channel x writing one to DMAC_CHER[x]. The DMA is ready and waiting for request.

8. Poll CBTC[x] bit in the DMAC_EBCISR.
9. If a new list of buffer shall be transferred repeat step 6. Check and handle HSMCI errors.
10. Poll FIFOEMPTY field in the HSMCI_SR.
11. Send The STOP_TRANSMISSION command writing the HSMCI_ARG then the HSMCI_CMDR.
12. Wait for XFRDONE in the HSMCI_SR.

33.8.8.2 Block Length is Not Multiple of 4 (HSMCI_DMA.ROPT = 0)

Two DMA Transfer descriptors are used to perform the HSMCI block transfer.

1. Use the previous step to configure the HSMCI to perform a READ_MULTIPLE_BLOCK command.
2. Issue a READ_MULTIPLE_BLOCK command.
3. Program the DMA Controller to use a list of descriptors.

33.14.15 HSMCI Interrupt Mask Register

Name: HSMCI_IMR

Address: 0xF000804C (0), 0xF000C04C (1)

Access: Read-only

31	30	29	28	27	26	25	24
UNRE	OVRE	ACKRCVE	ACKRCV	XFRDONE	FIFOEMPTY	DMADONE	BLKOVRE
23	22	21	20	19	18	17	16
CSTOE	DTOE	DCRCE	RTOE	RENDE	RCRCE	RDIRE	RINDE
15	14	13	12	11	10	9	8
–	–	CSRCV	SDIOWAIT	–	–	–	SDIOIRQA
7	6	5	4	3	2	1	0
–	–	NOTBUSY	DTIP	BLKE	TXRDY	RXRDY	CMDRDY

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

0: The corresponding interrupt is not enabled.

1: The corresponding interrupt is enabled.

- **CMDRDY: Command Ready Interrupt Mask**
- **RXRDY: Receiver Ready Interrupt Mask**
- **TXRDY: Transmit Ready Interrupt Mask**
- **BLKE: Data Block Ended Interrupt Mask**
- **DTIP: Data Transfer in Progress Interrupt Mask**
- **NOTBUSY: Data Not Busy Interrupt Mask**
- **SDIOIRQA: SDIO Interrupt for Slot A Interrupt Mask**
- **SDIOWAIT: SDIO Read Wait Operation Status Interrupt Mask**
- **CSRCV: Completion Signal Received Interrupt Mask**
- **RINDE: Response Index Error Interrupt Mask**
- **RDIRE: Response Direction Error Interrupt Mask**
- **RCRCE: Response CRC Error Interrupt Mask**
- **RENDE: Response End Bit Error Interrupt Mask**
- **RTOE: Response Time-out Error Interrupt Mask**
- **DCRCE: Data CRC Error Interrupt Mask**
- **DTOE: Data Time-out Error Interrupt Mask**
- **CSTOE: Completion Signal Time-out Error Interrupt Mask**

35.5 Product Dependencies

35.5.1 I/O Lines

The pins used for interfacing the compliant external devices may be multiplexed with PIO lines. The programmer must first program the PIO controllers to assign the TC pins to their peripheral functions.

Table 35-4. I/O Lines

Instance	Signal	I/O Line	Peripheral
TC0	TCLK0	PA24	A
TC0	TCLK1	PA25	A
TC0	TCLK2	PA26	A
TC0	TIOA0	PA21	A
TC0	TIOA1	PA22	A
TC0	TIOA2	PA23	A
TC0	TIOB0	PA27	A
TC0	TIOB1	PA28	A
TC0	TIOB2	PA29	A
TC1	TCLK3	PC4	C
TC1	TCLK4	PC7	C
TC1	TCLK5	PC14	C
TC1	TIOA3	PC2	C
TC1	TIOA4	PC5	C
TC1	TIOA5	PC12	C
TC1	TIOB3	PC3	C
TC1	TIOB4	PC6	C
TC1	TIOB5	PC13	C

35.5.2 Power Management

The TC is clocked through the Power Management Controller (PMC), thus the programmer must first configure the PMC to enable the Timer Counter clock.

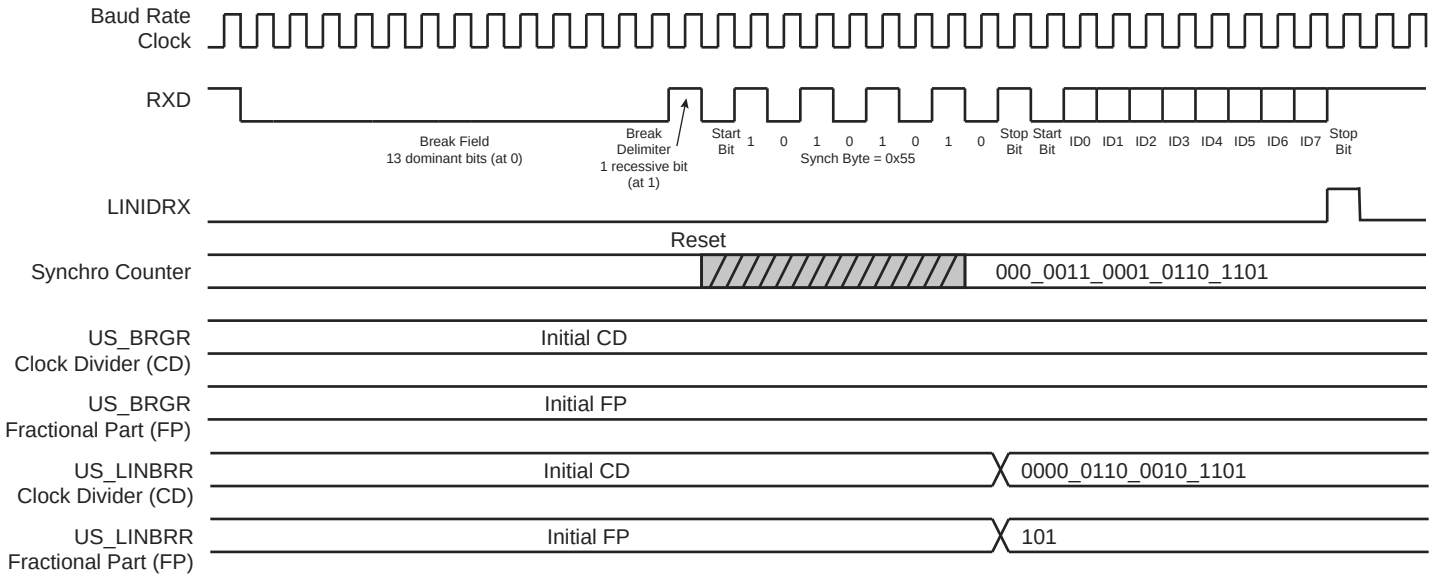
35.5.3 Interrupt Sources

The TC has an interrupt line connected to the interrupt controller. Handling the TC interrupt requires programming the interrupt controller before configuring the TC.

Table 35-5. Peripheral IDs

Instance	ID
TC0	17
TC1	17

Figure 38-42. Slave Node Synchronization



The accuracy of the synchronization depends on several parameters:

- Nominal clock frequency (f_{Nom}) (the theoretical slave node clock frequency)
- Baud Rate
- Oversampling (OVER = 0 => 16X or OVER = 0 => 8X)

The following formula is used to compute the deviation of the slave bit rate relative to the master bit rate after synchronization (f_{SLAVE} is the real slave node clock frequency):

$$\text{Baudrate\_deviation} = \left(100 \times \frac{[\alpha \times 8 \times (2 - \text{OVER}) + \beta] \times \text{Baudrate}}{8 \times f_{SLAVE}}\right)\%$$

$$\text{Baudrate\_deviation} = \left(100 \times \frac{[\alpha \times 8 \times (2 - \text{OVER}) + \beta] \times \text{Baudrate}}{8 \times \left(\frac{f_{TOL\_UNSYNCH}}{100}\right) \times f_{Nom}}\right)\%$$

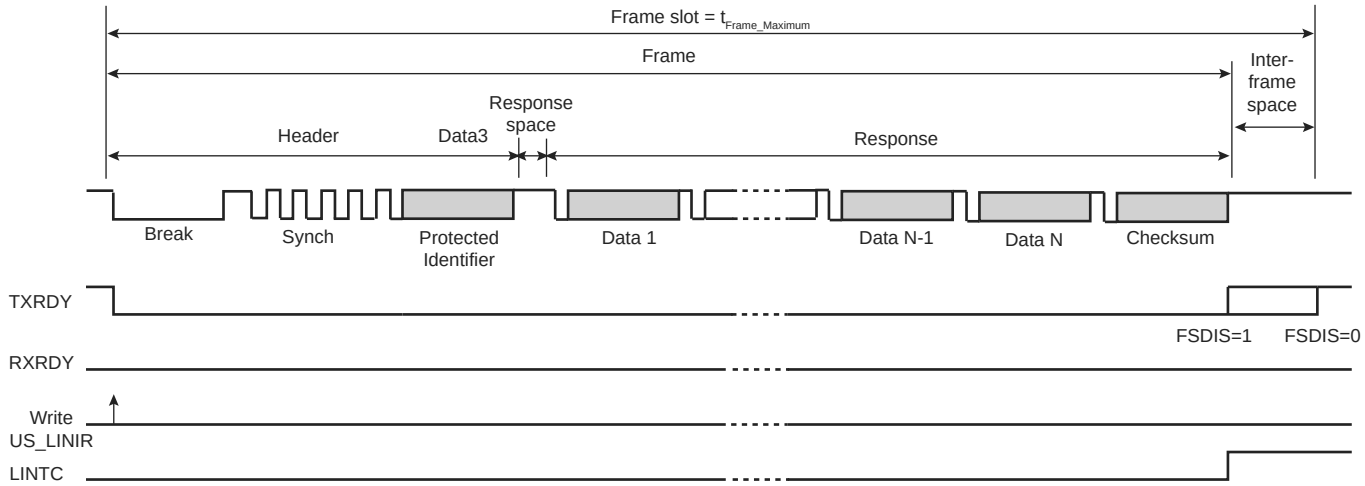
$$-0.5 \leq \alpha \leq +0.5 \quad -1 < \beta < +1$$

$f_{TOL\_UNSYNCH}$ is the deviation of the real slave node clock from the nominal clock frequency. The LIN Standard imposes that it must not exceed ±15%. The LIN Standard imposes also that for communication between two nodes, their bit rate must not differ by more than ±2%. This means that the baudrate_deviation must not exceed ±1%.

It follows from that, a minimum value for the nominal clock frequency:

$$f_{Nom}(\min) = \left(100 \times \frac{[0.5 \times 8 \times (2 - \text{OVER}) + 1] \times \text{Baudrate}}{8 \times \left(\frac{-15}{100} + 1\right) \times 1\%}\right)\text{Hz}$$

Figure 38-47. Master Node Configuration, NACT = IGNORE



Slave Node Configuration

- Write TXEN and RXEN in US_CR to enable both the transmitter and the receiver.
- Write USART_MODE in US_MR to select the LIN mode and the slave node configuration.
- Write CD and FP in US_BRGR to configure the baud rate.
- Wait until LINID in US_CSR rises
- Check LINISFE and LINPE errors
- Read IDCHR in US_RHR
- Write NACT, PARDIS, CHKDIS, CHKTYPE, DLCM and DLC in US_LINMR to configure the frame transfer.

IMPORTANT: If the NACT configuration for this frame is PUBLISH, the US_LINMR must be written with NACT = PUBLISH even if this field is already correctly configured, in order to set the TXREADY flag and the corresponding write transfer request.

What comes next depends on the NACT configuration:

- Case 1: NACT = PUBLISH, the LIN controller sends the response
  - Wait until TXRDY in US_CSR rises
  - Write TCHR in US_THR to send a byte
  - If all the data have not been written, redo the two previous steps
  - Wait until LINTC in US_CSR rises
  - Check the LIN errors
- Case 2: NACT = SUBSCRIBE, the USART receives the response
  - Wait until RXRDY in US_CSR rises
  - Read RCHR in US_RHR
  - If all the data have not been read, redo the two previous steps
  - Wait until LINTC in US_CSR rises
  - Check the LIN errors
- Case 3: NACT = IGNORE, the USART is not concerned by the response
  - Wait until LINTC in US_CSR rises
  - Check the LIN errors

39.6.4 UART Interrupt Disable Register

Name: UART_IDR

Address: 0xF804000C (0), 0xF804400C (1)

Access: Write-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
–	–	–	–	–	–	TXEMPTY	–
7	6	5	4	3	2	1	0
PARE	FRAME	OVRE	–	–	–	TXRDY	RXRDY

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

0: No effect.

1: Disables the corresponding interrupt.

- **RXRDY: Disable RXRDY Interrupt**
- **TXRDY: Disable TXRDY Interrupt**
- **OVRE: Disable Overrun Error Interrupt**
- **FRAME: Disable Framing Error Interrupt**
- **PARE: Disable Parity Error Interrupt**
- **TXEMPTY: Disable TXEMPTY Interrupt**

41.4 Software Modem Device (SMD) User Interface

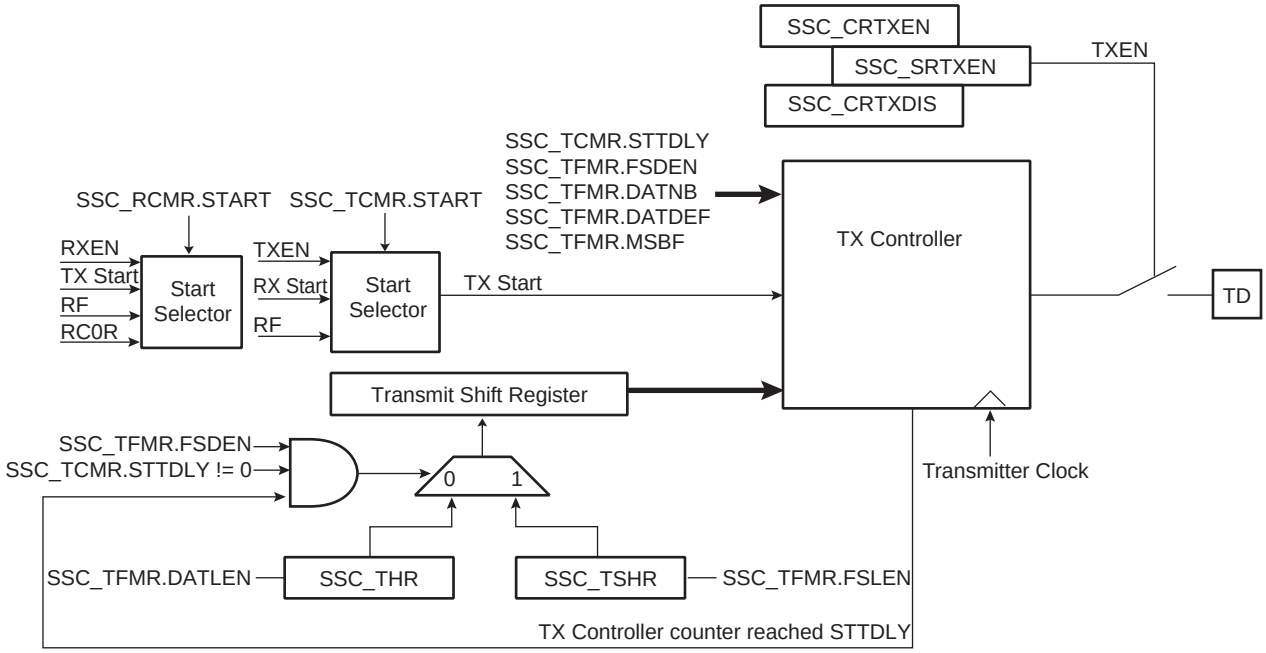
The SMD presents a number of registers through the AHB interface for software control and status functions.

Table 41-1. Register Mapping

Offset	Register	Name	Access	Reset
0x0C	SMD Drive register	SMD_DRIVE	Read/Write	0x00000002

When both the SSC_THR and the transmit shift register are empty, the status flag TXEMPTY is set in the SSC_SR. When the Transmit Holding register is transferred in the transmit shift register, the status flag TXRDY is set in the SSC_SR and additional data can be loaded in the holding register.

Figure 42-11. Transmitter Block Diagram



42.8.3 Receiver Operations

A received frame is triggered by a start event and can be followed by synchronization data before data transmission.

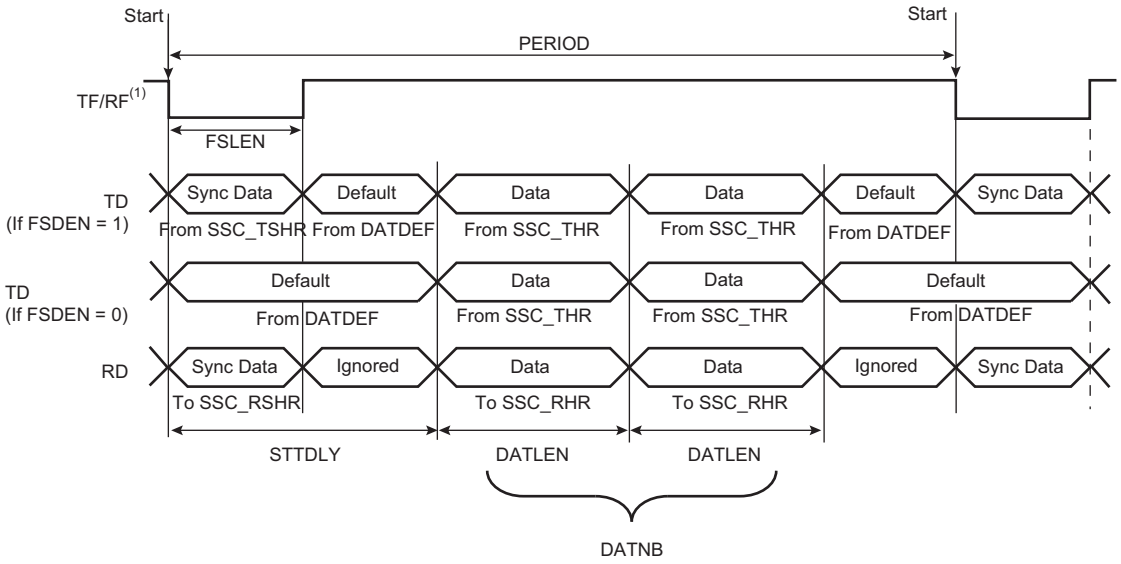
The start event is configured setting the Receive Clock Mode Register (SSC_RCMR). See Section 42.8.4 “Start” on page 969.

The frame synchronization is configured setting the Receive Frame Mode Register (SSC_RFMR). See Section 42.8.5 “Frame Sync” on page 971.

The receiver uses a shift register clocked by the receiver clock signal and the start mode selected in the SSC_RCMR. The data is transferred from the shift register depending on the data format selected.

When the receiver shift register is full, the SSC transfers this data in the holding register, the status flag RXRDY is set in the SSC_SR and the data can be read in the receiver holding register. If another transfer occurs before read of the Receive Holding Register (SSC_RHR), the status flag OVERUN is set in the SSC_SR and the receiver shift register is transferred in the SSC_RHR.

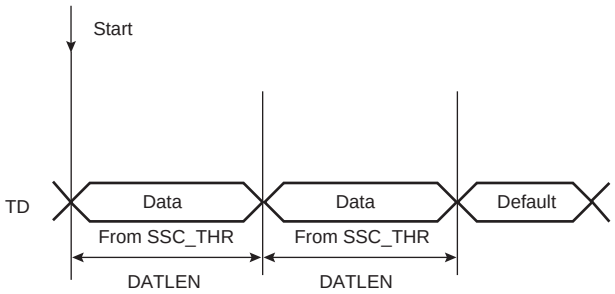
Figure 42-16. Transmit and Receive Frame Format in Edge/Pulse Start Modes



Note: 1. Example of input on falling edge of TF/RF.

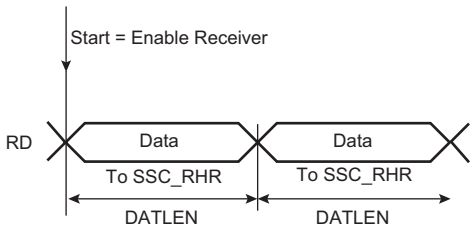
In the example illustrated in Figure 42-17 “Transmit Frame Format in Continuous Mode (STTDLY = 0)”, the SSC_THR is loaded twice. The FSDEN value has no effect on the transmission. SyncData cannot be output in continuous mode.

Figure 42-17. Transmit Frame Format in Continuous Mode (STTDLY = 0)



Start: 1. TXEMPTY set to 1
2. Write into the SSC_THR

Figure 42-18. Receive Frame Format in Continuous Mode (STTDLY = 0)



43.6 Image Sensor Interface (ISI) User Interface

Table 43-13. Register Mapping

Offset	Register	Name	Access	Reset Value
0x00	ISI Configuration 1 Register	ISI_CFG1	Read/Write	0x00000000
0x04	ISI Configuration 2 Register	ISI_CFG2	Read/Write	0x00000000
0x08	ISI Preview Size Register	ISI_PSIZE	Read/Write	0x00000000
0x0C	ISI Preview Decimation Factor Register	ISI_PDECF	Read/Write	0x00000010
0x10	ISI Color Space Conversion YCrCb To RGB Set 0 Register	ISI_Y2R_SET0	Read/Write	0x6832CC95
0x14	ISI Color Space Conversion YCrCb To RGB Set 1 Register	ISI_Y2R_SET1	Read/Write	0x00007102
0x18	ISI Color Space Conversion RGB To YCrCb Set 0 Register	ISI_R2Y_SET0	Read/Write	0x01324145
0x1C	ISI Color Space Conversion RGB To YCrCb Set 1 Register	ISI_R2Y_SET1	Read/Write	0x01245E38
0x20	ISI Color Space Conversion RGB To YCrCb Set 2 Register	ISI_R2Y_SET2	Read/Write	0x01384A4B
0x24	ISI Control Register	ISI_CR	Write-only	–
0x28	ISI Status Register	ISI_SR	Read-only	0x00000000
0x2C	ISI Interrupt Enable Register	ISI_IER	Write-only	–
0x30	ISI Interrupt Disable Register	ISI_IDR	Write-only	–
0x34	ISI Interrupt Mask Register	ISI_IMR	Read-only	0x00000000
0x38	DMA Channel Enable Register	ISI_DMA_CHER	Write-only	–
0x3C	DMA Channel Disable Register	ISI_DMA_CHDR	Write-only	–
0x40	DMA Channel Status Register	ISI_DMA_CHSR	Read-only	0x00000000
0x44	DMA Preview Base Address Register	ISI_DMA_P_ADDR	Read/Write	0x00000000
0x48	DMA Preview Control Register	ISI_DMA_P_CTRL	Read/Write	0x00000000
0x4C	DMA Preview Descriptor Address Register	ISI_DMA_P_DSCR	Read/Write	0x00000000
0x50	DMA Codec Base Address Register	ISI_DMA_C_ADDR	Read/Write	0x00000000
0x54	DMA Codec Control Register	ISI_DMA_C_CTRL	Read/Write	0x00000000
0x58	DMA Codec Descriptor Address Register	ISI_DMA_C_DSCR	Read/Write	0x00000000
0x5C–0xE0	Reserved	–	–	–
0xE4	Write Protection Mode Register	ISI_WPMR	Read/Write	0x00000000
0xE8	Write Protection Status Register	ISI_WPSR	Read-only	0x00000000
0xEC–0xF8	Reserved	–	–	–
0xFC	Reserved	–	–	–

Note: Several parts of the ISI controller use the pixel clock provided by the image sensor (ISI_PCK). Thus the user must first program the image sensor to provide this clock (ISI_PCK) before programming the Image Sensor Controller.