Microchip Technology - ATSAMS70N21A-CNT Datasheet

What is "Embedded - Microcontrollers"?

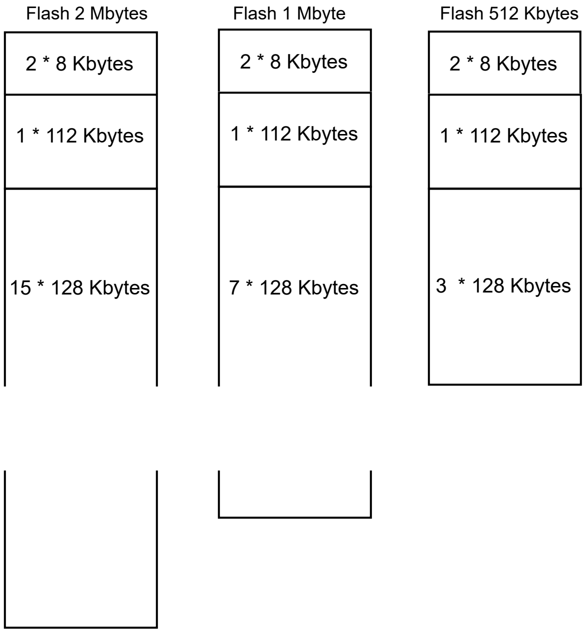
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	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	2MB (2M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	384K x 8
Voltage - Supply (Vcc/Vdd)	1.08V ~ 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/atsams70n21a-cnt

The figure below illustrates the organization of the Flash depending on its size.

Figure 11-3. Flash Size



Erasing the memory can be performed:

- by block of 8 Kbytes
- by sector of 128 Kbytes
- by 512-byte page for up to 8 Kbytes within a specific small sector
- Chip Erase

The memory has one additional reprogrammable page that can be used as page signature by the user. It is accessible through specific modes, for erase, write and read operations. Erase pin assertion will not erase the User Signature page.

Erase memory by page is possible only in a sector of 8 Kbytes.

EWP and EWPL commands can be only used in 8-Kbyte sectors.

11.1.5.2 Enhanced Embedded Flash Controller

Each Enhanced Embedded Flash Controller manages accesses performed by the masters of the system. It enables reading the Flash and writing the write buffer. It also contains a User Interface, mapped on the APB.

The Enhanced Embedded Flash Controller ensures the interface of the Flash block.

It manages the programming, erasing, locking and unlocking sequences of the Flash using a full set of commands.

One of the commands returns the embedded Flash descriptor definition that informs the system about the Flash organization, thus making the software generic.

11.1.5.3 Flash Speed

The user must set the number of wait states depending on the system frequency.

For more details, refer to Embedded Flash Characteristics.

18. Fast Flash Programming Interface (FFPI)

18.1 Description

The Fast Flash Programming Interface (FFPI) provides parallel high-volume programming using a standard gang programmer. The parallel interface is fully handshaked and the device is considered to be a standard EEPROM. Additionally, the parallel protocol offers an optimized access to all the embedded Flash functionalities.

Although the Fast Flash Programming mode is a dedicated mode for high volume programming, this mode is not designed for in-situ programming.

18.2 Embedded Characteristics

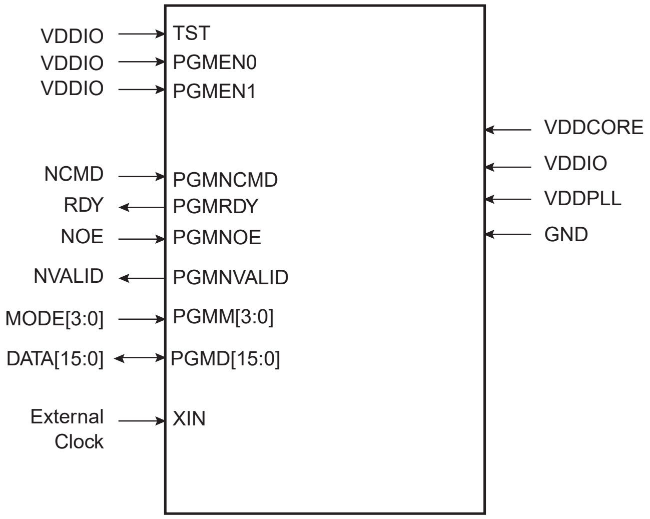
- Programming Mode for High-volume Flash Programming Using Gang Programmer
  - Offers Read and Write Access to the Flash Memory Plane
  - Enables Control of Lock Bits and General-purpose NVM Bits
  - Enables Security Bit Activation
  - Disabled Once Security Bit is Set
- Parallel Fast Flash Programming Interface
  - Provides a 16-bit Parallel Interface to Program the Embedded Flash
  - Full Handshake Protocol

18.3 Parallel Fast Flash Programming

18.3.1 Device Configuration

In Fast Flash Programming mode, the device is in a specific test mode. Only a certain set of pins is significant. The rest of the PIOs are used as inputs with a pullup. The crystal oscillator is in Bypass mode. Other pins must be left unconnected.

Figure 18-1. 16-bit Parallel Programming Interface



Bit 16 – UECCELSB Unique ECC Error on LSB Part of the Memory Flash Data Bus (cleared on read)

Value	Description
0	No unique error detected on 64 LSB data bus of the Flash memory since the last read of EEFC_FSR.
1	One unique error detected but corrected on 64 LSB data bus of the Flash memory since the last read of EEFC_FSR.

Bit 3 – FLERR Flash Error Status (cleared when a programming operation starts)

Value	Description
0	No Flash memory error occurred at the end of programming (EraseVerify or WriteVerify test has passed).
1	A Flash memory error occurred at the end of programming (EraseVerify or WriteVerify test has failed).

Bit 2 – FLOCKE Flash Lock Error Status (cleared on read)
This flag is automatically cleared when EEFC_FSR is read or EEFC_FCR is written.

Value	Description
0	No programming/erase of at least one locked region has happened since the last read of EEFC_FSR.
1	Programming/erase of at least one locked region has happened since the last read of EEFC_FSR.

Bit 1 – FCMDE Flash Command Error Status (cleared on read or by writing EEFC_FCR)

Value	Description
0	No invalid commands and no bad keywords were written in EEFC_FMR.
1	An invalid command and/or a bad keyword was/were written in EEFC_FMR.

Bit 0 – FRDY Flash Ready Status (cleared when Flash is busy)
When set, this flag triggers an interrupt if the FRDY flag is set in EEFC_FMR.

This flag is automatically cleared when the EEFC is busy.

Value	Description
0	The EEFC is busy.
1	The EEFC is ready to start a new command.

38.8.9 GMAC Receive Status Register

Name: GMAC_RSR
Offset: 0x020
Reset: 0x00000000
Property: -

This register, when read, provides receive status details. Once read, individual bits may be cleared by writing a '1' to them. It is not possible to set a bit to '1' by writing to this register.

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
					HNO	RXOVR	REC	BNA
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Bit 3 – HNO HRESP Not OK
This bit is set when the DMA block sees HRESP not OK.

This bit is cleared by writing a '1' to it.

Bit 2 – RXOVR Receive Overrun
This bit is set if the receive status was not taken at the end of the frame. The buffer will be recovered if an overrun occurs.

This bit is cleared by writing a '1' to it.

Bit 1 – REC Frame Received
This bit is set to when one or more frames have been received and placed in memory.

This bit is cleared by writing a '1' to it.

Bit 0 – BNA Buffer Not Available
When this bit is set, an attempt was made to get a new buffer and the pointer indicated that it was owned by the processor. The DMA will re-read the pointer each time an end of frame is received until a valid pointer is found. This bit is set following each descriptor read attempt that fails, even if consecutive pointers are unsuccessful and software has in the mean time cleared the status flag.

39.6 Register Summary

Offset	Name	Bit Pos.								
0x00	USBHS_DEVCTRL	7:0	ADDEN	UADD[6:0]						
		15:8	TSTPCKT	TSTK	TSTJ	LS	SPDCONF[1:0]		RMWKUP	DETACH
		23:16								OPMODE2
		31:24								
0x04	USBHS_DEVISR	7:0		UPRSM	EORSM	WAKEUP	EORST	SOF	MSOF	SUSP
		15:8	PEP_3	PEP_2	PEP_1	PEP_0				
		23:16			PEP_9	PEP_8	PEP_7	PEP_6	PEP_5	PEP_4
		31:24	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	DMA_0	
0x08	USBHS_DEVICR	7:0		UPRSMC	EORSMC	WAKEUPC	EORSTC	SOFC	MSOFC	SUSPC
		15:8								
		23:16								
		31:24								
0x0C	USBHS_DEVIFR	7:0		UPRSMS	EORSMS	WAKEUPS	EORSTS	SOFS	MSOFS	SUSPS
		15:8								
		23:16								
		31:24	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	DMA_0	
0x10	USBHS_DEVIMR	7:0		UPRSME	EORSME	WAKEUPE	EORSTE	SOFE	MSOFE	SUSPE
		15:8	PEP_3	PEP_2	PEP_1	PEP_0				
		23:16			PEP_9	PEP_8	PEP_7	PEP_6	PEP_5	PEP_4
		31:24	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	DMA_0	
0x14	USBHS_DEVIDR	7:0		UPRSMEC	EORSMEC	WAKEUPEC	EORSTEC	SOFEC	MSOFEC	SUSPEC
		15:8	PEP_3	PEP_2	PEP_1	PEP_0				
		23:16			PEP_9	PEP_8	PEP_7	PEP_6	PEP_5	PEP_4
		31:24	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	DMA_0	
0x18	USBHS_DEVIER	7:0		UPRSMES	EORSMES	WAKEUPES	EORSTES	SOFES	MSOFES	SUSPES
		15:8	PEP_3	PEP_2	PEP_1	PEP_0				
		23:16			PEP_9	PEP_8	PEP_7	PEP_6	PEP_5	PEP_4
		31:24	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	DMA_0	
0x1C	USBHS_DEVEPT	7:0	EPEN7	EPEN6	EPEN5	EPEN4	EPEN3	EPEN2	EPEN1	EPEN0
		15:8							EPEN9	EPEN8
		23:16	EPRST7	EPRST6	EPRST5	EPRST4	EPRST3	EPRST2	EPRST1	EPRST0
		31:24							EPRST9	EPRST8
0x20	USBHS_DEVFNUM	7:0	FNUM[4:0]					MFNUM[2:0]		
		15:8	FNCERR		FNUM[10:5]					
		23:16								
		31:24								
0x24 ... 0xFF	Reserved									
0x0100	USBHS_DEVEPTCFG0	7:0		EPSIZE[2:0]			EPBK[1:0]		ALLOC	
		15:8		NBTRANS[1:0]		EPTYPE[1:0]			AUTOSW	EPDIR
		23:16								
		31:24								

Offset	Name	Bit Pos.								
0x0644	USBHS_HSTPIPIDR9 (INTPIPES)	7:0	SHORTPACKETIEC	RXSTALLDEC	OVERFIEC	NAKEDEC	PERREC	UNDERFIEC	TXOUTEC	RXINEC
		15:8		FIFOCONC		NBUSYBKEC				
		23:16							PFREEZEC	PDISHDMAC
		31:24								
0x0644	USBHS_HSTPIPIDR9 (ISOPIPES)	7:0	SHORTPACKETIEC	CRCERREC	OVERFIEC	NAKEDEC	PERREC	UNDERFIEC	TXOUTEC	RXINEC
		15:8		FIFOCONC		NBUSYBKEC				
		23:16							PFREEZEC	PDISHDMAC
		31:24								
0x0648 ... 0x064F	Reserved									
0x0650	USBHS_HSTPIPINRQ0	7:0	INRQ[7:0]							
		15:8								INMODE
		23:16								
		31:24								
0x0654	USBHS_HSTPIPINRQ1	7:0	INRQ[7:0]							
		15:8								INMODE
		23:16								
		31:24								
0x0658	USBHS_HSTPIPINRQ2	7:0	INRQ[7:0]							
		15:8								INMODE
		23:16								
		31:24								
0x065C	USBHS_HSTPIPINRQ3	7:0	INRQ[7:0]							
		15:8								INMODE
		23:16								
		31:24								
0x0660	USBHS_HSTPIPINRQ4	7:0	INRQ[7:0]							
		15:8								INMODE
		23:16								
		31:24								
0x0664	USBHS_HSTPIPINRQ5	7:0	INRQ[7:0]							
		15:8								INMODE
		23:16								
		31:24								
0x0668	USBHS_HSTPIPINRQ6	7:0	INRQ[7:0]							
		15:8								INMODE
		23:16								
		31:24								
0x066C	USBHS_HSTPIPINRQ7	7:0	INRQ[7:0]							
		15:8								INMODE
		23:16								
		31:24								

39.6.14 Device Endpoint x Configuration Register

Name:	USBHS_DEVEPTCFGx
Offset:	0x0100 + x*0x04 [x=0..9]
Reset:	0
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
		NBTRANS[1:0]		EPTYPE[1:0]			AUTOSW	EPDIR
Access								
Reset		0	0	0	0		0	0

Bit	7	6	5	4	3	2	1	0
		EPSIZE[2:0]			EPBK[1:0]		ALLOC	
Access								
Reset		0	0	0	0	0	0	

Bits 14:13 – NBTRANS[1:0] Number of transactions per microframe for isochronous endpoint
This field should be written with the number of transactions per microframe to perform high-bandwidth isochronous transfer.

It can be written only for endpoints that have this capability (see USBHS_FEATURES.ENHBISOx bit). Otherwise, this field is 0.

This field is irrelevant for non-isochronous endpoints.

Value	Name	Description
0	0_TRANS	Reserved to endpoint that does not have the high-bandwidth isochronous capability.
1	1_TRANS	Default value: one transaction per microframe.
2	2_TRANS	Two transactions per microframe. This endpoint should be configured as double-bank.
3	3_TRANS	Three transactions per microframe. This endpoint should be configured as triple-bank.

Bits 12:11 – EPTYPE[1:0] Endpoint Type
This field should be written to select the endpoint type:

This field is cleared upon receiving a USB reset.

39.6.33 Host Global Interrupt Clear Register

Name: USBHS_HSTICR
Offset: 0x0408
Property: Write-only

This register always reads as zero.

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

0: No effect.

1: Clears the corresponding bit in USBHS_HSTISR.

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
		HWUPIC	HSOFIC	RXRSMIC	RSMEDIC	RSTIC	DDISCIC	DCONNIC
Access								
Reset								

Bit 6 – HWUPIC Host Wakeup Interrupt Clear

Bit 5 – HSOFIC Host Start of Frame Interrupt Clear

Bit 4 – RXRSMIC Upstream Resume Received Interrupt Clear

Bit 3 – RSMEDIC Downstream Resume Sent Interrupt Clear

Bit 2 – RSTIC USB Reset Sent Interrupt Clear

Bit 1 – DDISCIC Device Disconnection Interrupt Clear

Bit 0 – DCONNIC Device Connection Interrupt Clear

39.6.35 Host Global Interrupt Mask Register

Name: USBHS_HSTIMR
Offset: 0x0410
Reset: 0x00000000
Property: Read-only

Bit	31	30	29	28	27	26	25	24
	DMA_6	DMA_5	DMA_4	DMA_3	DMA_2	DMA_1	DMA_0	
Access								
Reset	0	0	0	0	0	0	0	

Bit	23	22	21	20	19	18	17	16
							PEP_9	PEP_8
Access								
Reset							0	0

Bit	15	14	13	12	11	10	9	8
	PEP_7	PEP_6	PEP_5	PEP_4	PEP_3	PEP_2	PEP_1	PEP_0
Access								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
		HWUPIE	HSOFIE	RXRSMIE	RSMEDIE	RSTIE	DDISCIE	DCONNIE
Access								
Reset		0	0	0	0	0	0	0

Bits 25, 26, 27, 28, 29, 30, 31 – DMA_ DMA Channel x Interrupt Enable

Value	Description
0	Cleared when the corresponding bit in USBHS_HSTIDR = 1. This disables the DMA Channel x Interrupt (USBHS_HSTISR.DMA_x).
1	Set when the corresponding bit in USBHS_HSTIER = 1. This enables the DMA Channel x Interrupt (USBHS_HSTISR.DMA_x).

Bits 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 – PEP_ Pipe x Interrupt Enable

Value	Description
0	Cleared when PEP_x = 1. This disables the Pipe x Interrupt (PEP_x).
1	Set when the corresponding bit in USBHS_HSTIER = 1. This enables the Pipe x Interrupt (USBHS_HSTISR.PEP_x).

Bit 6 – HWUPIE Host Wakeup Interrupt Enable

Value	Description
0	Cleared when USBHS_HSTIDR.HWUPIEC = 1. This disables the Host Wakeup Interrupt (USBHS_HSTISR.HWUPI).
1	Set when USBHS_HSTIER.HWUPIES = 1. This enables the Host Wakeup Interrupt (USBHS_HSTISR.HWUPI).

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 RXSTALLDI or PERRI = 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

44.8.3 Receive Operations

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

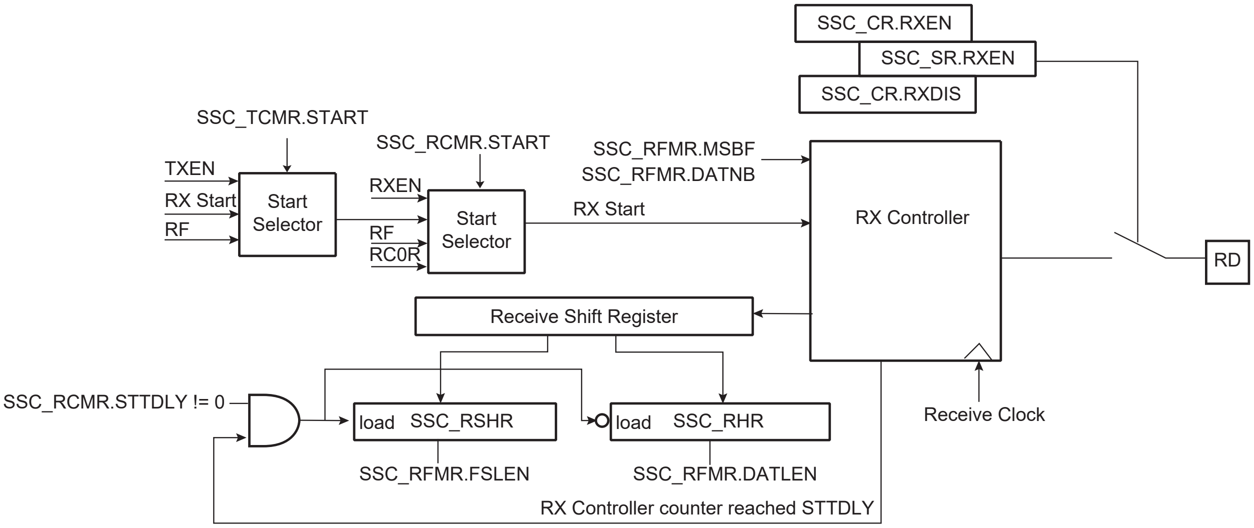
The start event is configured setting the Receive Clock Mode Register (SSC_RCMR). See Start.

The frame synchronization is configured by setting the Receive Frame Mode Register (SSC_RFMR). See Frame Synchronization.

The receiver uses a shift register clocked by the receive 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 44-12. Receive Block Diagram



44.8.4 Start

The transmitter and receiver can both be programmed to start their operations when an event occurs, respectively in the Transmit Start Selection (START) field of SSC_TCMR and in the Receive Start Selection (START) field of SSC_RCMR.

Under the following conditions the start event is independently programmable:

- Continuous. In this case, the transmission starts as soon as a word is written in SSC_THR and the reception starts as soon as the receiver is enabled.
- Synchronously with the transmitter/receiver
- On detection of a falling/rising edge on TF/RF
- On detection of a low level/high level on TF/RF
- On detection of a level change or an edge on TF/RF

A start can be programmed in the same manner on either side of the Transmit/Receive Clock Register (SSC_RCMR/SSC_TCMR). Thus, the start could be on TF (Transmit) or RF (Receive).

Moreover, the receiver can start when data is detected in the bit stream with the Compare Functions.

47.6.2 UART Mode Register

Name: UART_MR
Offset: 0x04
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
	CHMODE[1:0]			BRSRCCK	PAR[2:0]			
Access	R/W	R/W		R/W	R/W	R/W	R/W	
Reset	0	0		0	0	0	0	

Bit	7	6	5	4	3	2	1	0
				FILTER				
Access				R/W				
Reset				0				

Bits 15:14 – CHMODE[1:0] Channel Mode

Value	Name	Description
0	NORMAL	Normal mode
1	AUTOMATIC	Automatic echo
2	LOCAL_LOOPBACK	Local loopback
3	REMOTE_LOOPBACK	Remote loopback

Bit 12 – BRSRCCK Baud Rate Source Clock
0 (PERIPH_CLK): The baud rate is driven by the peripheral clock

1 (PMC_PCK): The baud rate is driven by a PMC-programmable clock PCK (see section "Power Management Controller (PMC)").

Bits 11:9 – PAR[2:0] Parity Type

Value	Name	Description
0	EVEN	Even Parity
1	ODD	Odd Parity
2	SPACE	Space: parity forced to 0
3	MARK	Mark: parity forced to 1
4	NO	No parity

SFID2[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 a Rx buffer

1: Debug Message A

2: Debug Message B

3: Debug Message C

SFID2[5:0] defines the index of the dedicated Rx Buffer element to which a matching message is stored.

49.5.7.6 Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, its address is the Filter List Extended Start Address MCAN_XIDFC.FLESA plus two times the index of the filter element (0…63).

Table 49-11. Extended Message ID Filter Element

	31			24	23	16	15	8	7	0
F0	EFEC [2:0]			EFID1[28:0]						
F1	EFT[1:0]		–	EFID2[28:0]						

• F0 Bit 31:29 EFEC[2:0]: Extended Filter Element Configuration

All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = “100”, “101”, or “110”, a match sets the interrupt flag MCAN_IR.HPM and, if enabled, an interrupt is generated. In this case, register MCAN_HPMS is updated with the status of the priority match.

0: Disable filter element

1: Store in Rx FIFO 0 if filter matches

2: Store in Rx FIFO 1 if filter matches

3: Reject ID if filter matches

4: Set priority if filter matches

5: Set priority and store in FIFO 0 if filter matches

6: Set priority and store in FIFO 1 if filter matches

7: Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored

• F0 Bits 28:0 EFID1[28:0]: Extended Filter ID 1

First ID of extended ID filter element.

When filtering for Rx Buffers or for debug messages this field defines the ID of an extended message to be stored. The received identifiers must match exactly, only MCAN_XIDAM masking mechanism (see Extended Message ID Filtering) is used.

• F1 Bits 31:30 EFT[1:0]: Extended Filter Type

0: Range filter from EF1ID to EF2ID (EF2ID ≥ EF1ID)

49.6.1 MCAN Core Release Register

Name: MCAN_CREL
Offset: 0x00
Reset: 0xrrrddddd
Property: Read-only

Due to clock domain crossing, there is a delay between when a register bit or field is written and when the related status register bits are updated.

Bit	31	30	29	28	27	26	25	24
	REL[3:0]				STEP[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bit	23	22	21	20	19	18	17	16
	SUBSTEP[3:0]				YEAR[3:0]			
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

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

Bit	7	6	5	4	3	2	1	0
	DAY[7:0]							
Access	R	R	R	R	R	R	R	R
Reset	x	x	x	x	x	x	x	x

Bits 31:28 – REL[3:0] Core Release
One digit, BCD-coded.

Bits 27:24 – STEP[3:0] Step of Core Release
One digit, BCD-coded.

Bits 23:20 – SUBSTEP[3:0] Sub-step of Core Release
One digit, BCD-coded.

Bits 19:16 – YEAR[3:0] Timestamp Year
One digit, BCD-coded. This field is set by generic parameter on MCAN synthesis.

Bits 15:8 – MON[7:0] Timestamp Month
Two digits, BCD-coded. This field is set by generic parameter on MCAN synthesis.

Bits 7:0 – DAY[7:0] Timestamp Day
Two digits, BCD-coded. This field is set by generic parameter on MCAN synthesis.

49.6.35 MCAN Tx Buffer Configuration

Name: MCAN_TXBC
Offset: 0xC0
Reset: 0x00000000
Property: Read/Write

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

The sum of TFQS and NDTB may not exceed 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers.

Bit	31	30	29	28	27	26	25	24
		TFQM	TFQS[5:0]					
Access		R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset		0	0	0	0	0	0	0

Bit	23	22	21	20	19	18	17	16
			NDTB[5:0]					
Access			R/W	R/W	R/W	R/W	R/W	R/W
Reset			0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
	TBSA[13:6]							
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
	TBSA[5:0]							
Access	R/W	R/W	R/W	R/W	R/W	R/W		
Reset	0	0	0	0	0	0		

Bit 30 – TFQM Tx FIFO/Queue Mode

Value	Description
0	Tx FIFO operation.
1	Tx Queue operation.

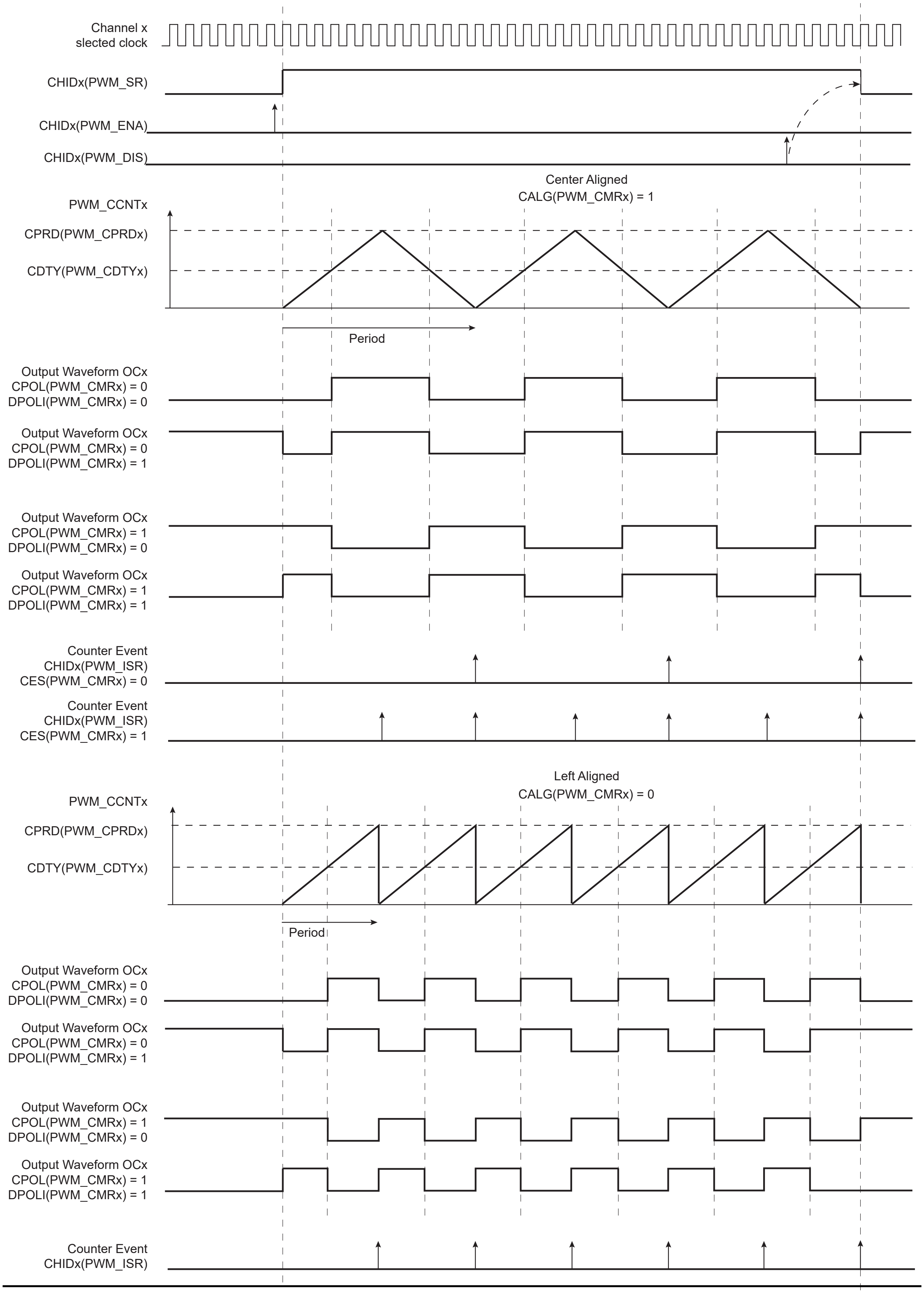
Bits 29:24 – TFQS[5:0] Transmit FIFO/Queue Size

Value	Description
0	No Tx FIFO/Queue.
1-32	Number of Tx Buffers used for Tx FIFO/Queue.
>32	Values greater than 32 are interpreted as 32.

Bits 21:16 – NDTB[5:0] Number of Dedicated Transmit Buffers

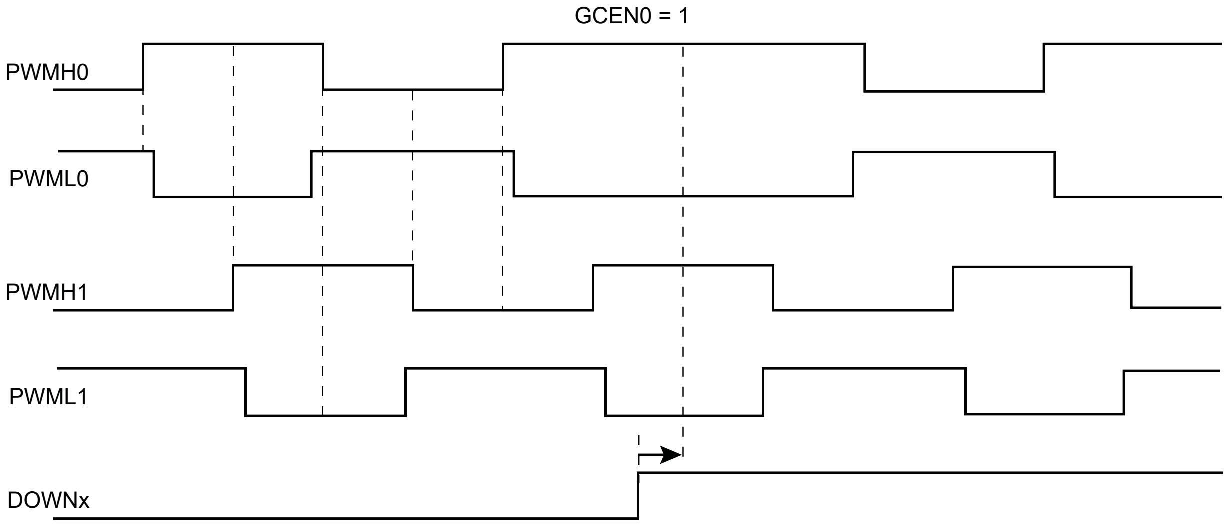
Value	Description
0	No dedicated Tx Buffers.
1-32	Number of dedicated Tx Buffers.
>32	Values greater than 32 are interpreted as 32.

Figure 51-5. Waveform Properties



When GCEN0 is set to '1', channels 0 and 1 outputs are driven with gray counter.

Figure 51-8. 2-bit Gray Up/Down Counter



51.6.2.5 Dead-Time Generator

The dead-time generator uses the comparator output OCx to provide the two complementary outputs DTOHx and DTOLx, which allows the PWM macrocell to drive external power control switches safely. When the dead-time generator is enabled by setting the bit DTE to 1 or 0 in the PWM Channel Mode Register (PWM_CMRx), dead-times (also called dead-bands or non-overlapping times) are inserted between the edges of the two complementary outputs DTOHx and DTOLx. Note that enabling or disabling the dead-time generator is allowed only if the channel is disabled.

The dead-time is adjustable by the PWM Channel Dead Time Register (PWM_DTx). Each output of the dead-time generator can be adjusted separately by DTH and DTL. The dead-time values can be updated synchronously to the PWM period by using the PWM Channel Dead Time Update Register (PWM_DTUPDx).

The dead-time is based on a specific counter which uses the same selected clock that feeds the channel counter of the comparator. Depending on the edge and the configuration of the dead-time, DTOHx and DTOLx are delayed until the counter has reached the value defined by DTH or DTL. An inverted configuration bit (DTHI and DTLI bit in PWM_CMRx) is provided for each output to invert the dead-time outputs. The following figure shows the waveform of the dead-time generator.

Offset	Name	Bit Pos.								
0x0248	PWM_CDTYUPD2	7:0	CDTYUPD[7:0]							
		15:8	CDTYUPD[15:8]							
		23:16	CDTYUPD[23:16]							
		31:24								
0x024C	PWM_CPRD2	7:0	CPRD[7:0]							
		15:8	CPRD[15:8]							
		23:16	CPRD[23:16]							
		31:24								
0x0250	PWM_CPRDUPD2	7:0	CPRDUPD[7:0]							
		15:8	CPRDUPD[15:8]							
		23:16	CPRDUPD[23:16]							
		31:24								
0x0254	PWM_CCNT2	7:0	CNT[7:0]							
		15:8	CNT[15:8]							
		23:16	CNT[23:16]							
		31:24								
0x0258	PWM_DT2	7:0	DTH[7:0]							
		15:8	DTH[15:8]							
		23:16	DTL[7:0]							
		31:24	DTL[15:8]							
0x025C	PWM_DTUPD2	7:0	DTHUPD[7:0]							
		15:8	DTHUPD[15:8]							
		23:16	DTLUPD[7:0]							
		31:24	DTLUPD[15:8]							
0x0260	PWM_CMR3	7:0					CPRE[3:0]			
		15:8			TCTS	DPOLI	UPDS	CES	CPOL	CALG
		23:16					PPM	DTLI	DTHI	DTE
		31:24								
0x0264	PWM_CDTY3	7:0	CDTY[7:0]							
		15:8	CDTY[15:8]							
		23:16	CDTY[23:16]							
		31:24								
0x0268	PWM_CDTYUPD3	7:0	CDTYUPD[7:0]							
		15:8	CDTYUPD[15:8]							
		23:16	CDTYUPD[23:16]							
		31:24								
0x026C	PWM_CPRD3	7:0	CPRD[7:0]							
		15:8	CPRD[15:8]							
		23:16	CPRD[23:16]							
		31:24								
0x0270	PWM_CPRDUPD3	7:0	CPRDUPD[7:0]							
		15:8	CPRDUPD[15:8]							
		23:16	CPRDUPD[23:16]							
		31:24								
0x0274	PWM_CCNT3	7:0	CNT[7:0]							
		15:8	CNT[15:8]							

51.7.42 PWM Channel Duty Cycle Update Register

Name:	PWM_CDTYUPDx
Offset:	0x0208 + x*0x20 [x=0..3]
Reset:	–
Property:	Write-only

This register acts as a double buffer for the CDTY value. This prevents an unexpected waveform when modifying the waveform duty-cycle.

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
	CDTYUPD[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
	CDTYUPD[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
	CDTYUPD[7:0]							
Access	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	–

Bits 23:0 – CDTYUPD[23:0] Channel Duty-Cycle Update
Defines the waveform duty-cycle. This value must be defined between 0 and CPRD (PWM_CPRDx).

Signal Name	Recommended Pin Connection	Description
Fast Flash Programming Interface		
PGMEN0–PGMEN1	To enter in FFPI mode TST pins must be tied to VDDIO.	Programming Enabling Pulled-up inputs (100 kOhm) to VDDIO at reset.
PGMM0–PGMM3	Application dependent.	Programming Mode Pulled-up inputs (100 kOhm) to VDDIO at reset.
PGMD0–PGMD15	Application dependent.	Programming Data Pulled-up inputs (100 kOhm) to VDDIO at reset.
PGMRDY	Application dependent.	Programming Ready Pulled-up input (100 kOhm) to VDDIO at reset.
PGMNVALID	Application dependent.	Data Direction Pulled-up input (100 kOhm) to VDDIO at reset.
PGMNOE	Application dependent.	Programming Read Pulled-up input (100 kOhm) to VDDIO at reset.
PGMNCMD	Application dependent.	Programming Command Pulled-up input (100 kOhm) to VDDIO at reset.

60.3 Boot Program Hardware Constraints

Refer to 17. SAM-BA Boot Program for more details on the boot program.

60.3.1 Boot Program Supported Crystals (MHz)

A 12 MHz or a 16 MHz crystal or external clock (in Bypass mode) is mandatory in order to generate USB and PLL clocks correctly for the following boots.

60.3.2 SAM-BA Boot

The SAM-BA Boot Assistant supports serial communication via the UART or USB device port:

- UART0 hardware requirements: None
- USB Device hardware requirements: Eexternal crystal or external clock (see **Note**) with a frequency of 12 MHz or 16 MHz

Note: Must be 2500 ppm and VDDIO square wave signal.

Table 60-1. Pins Driven During SAM-BA Boot Program Execution

Peripheral	Pin
UART0	URXD0
UART0	UTXD0