Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

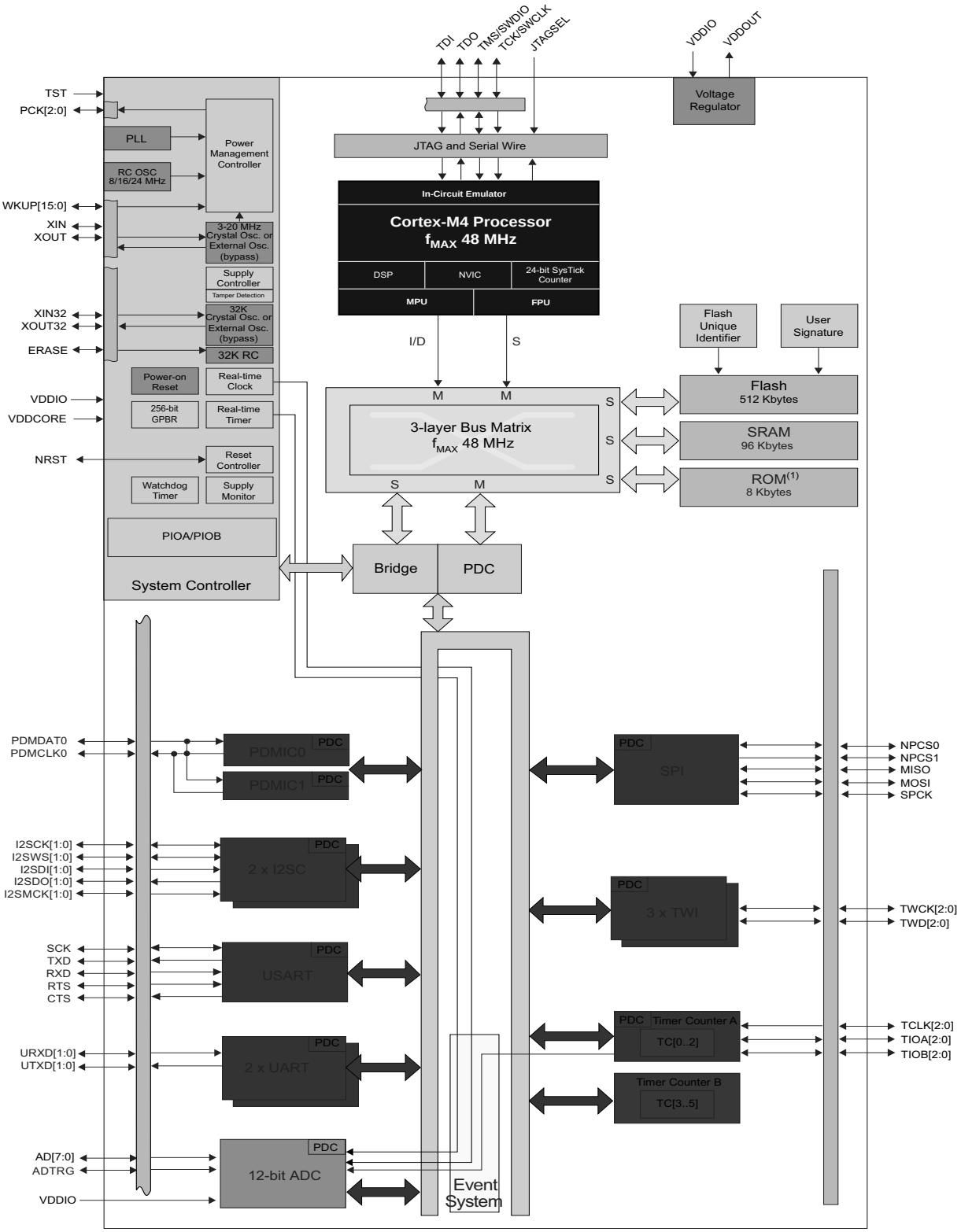
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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®-M4
Core Size	32-Bit Single-Core
Speed	48MHz
Connectivity	I²C, SPI, UART/USART
Peripherals	DMA, I²S, POR, PWM, WDT
Number of I/O	38
Program Memory Size	512KB (512K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	96K x 8
Voltage - Supply (Vcc/Vdd)	1.62V ~ 3.6V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atsamg53n19a-au

2. Block Diagram

Figure 2-1. SAM G53 Block Diagram



Note: 1. The ROM is reserved for future use.

11. Event System

The events generated by peripherals are designed to be directly routed to peripherals managing/using these events without processor intervention. Peripherals receiving events contain logic by which to select the one required.

11.1 Embedded Characteristics

- Timers, IO, peripherals generate event triggers which are directly routed to event managers such as ADC, to start measurement/conversion without processor intervention.
- UART, USART, SPI, TWI, ADC also generate event triggers directly connected to Peripheral DMA Controller (PDC) for data transfer without processor intervention.
- PMC security event (clock failure detection) can be programmed to switch the MCK on a reliable main RC internal clock without processor intervention.

12.6.5 General Data Processing Instructions

The table below shows the data processing instructions.

Table 12-20. Data Processing Instructions

Mnemonic	Description
ADC	Add with Carry
ADD	Add
ADDW	Add
AND	Logical AND
ASR	Arithmetic Shift Right
BIC	Bit Clear
CLZ	Count leading zeros
CMN	Compare Negative
CMP	Compare
EOR	Exclusive OR
LSL	Logical Shift Left
LSR	Logical Shift Right
MOV	Move
MOVT	Move Top
MOVW	Move 16-bit constant
MVN	Move NOT
ORN	Logical OR NOT
ORR	Logical OR
RBIT	Reverse Bits
REV	Reverse byte order in a word
REV16	Reverse byte order in each halfword
REVSH	Reverse byte order in bottom halfword and sign extend
ROR	Rotate Right
RRX	Rotate Right with Extend
RSB	Reverse Subtract
SADD16	Signed Add 16
SADD8	Signed Add 8
SASX	Signed Add and Subtract with Exchange
SSAX	Signed Subtract and Add with Exchange
SBC	Subtract with Carry
SHADD16	Signed Halving Add 16
SHADD8	Signed Halving Add 8
SHASX	Signed Halving Add and Subtract with Exchange
SHSAX	Signed Halving Subtract and Add with Exchange

Restrictions

Do not use SP and do not use PC.

Condition Flags

These instructions do not affect the condition code flags.

Examples

```
UQADD16  R7, R4, R2   ; Adds halfwords in R4 to corresponding halfword in R2,
                      ; saturates to 16 bits, writes to corresponding halfword of R7
UQADD8   R4, R2, R5   ; Adds bytes of R2 to corresponding byte of R5, saturates
                      ; to 8 bits, writes to corresponding bytes of R4
UQSUB16  R6, R3, R0   ; Subtracts halfwords in R0 from corresponding halfword
                      ; in R3, saturates to 16 bits, writes to corresponding
                      ; halfword in R6
UQSUB8   R1, R5, R6   ; Subtracts bytes in R6 from corresponding byte of R5,
                      ; saturates to 8 bits, writes to corresponding byte of R1.
```

12.6.8 Packing and Unpacking Instructions

The table below shows the instructions that operate on packing and unpacking data.

Table 12-23. Packing and Unpacking Instructions

Mnemonic	Description
PKH	Pack Halfword
SXTAB	Extend 8 bits to 32 and add
SXTAB16	Dual extend 8 bits to 16 and add
SXTAH	Extend 16 bits to 32 and add
SXTB	Sign extend a byte
SXTB16	Dual extend 8 bits to 16 and add
SXTH	Sign extend a halfword
UXTAB	Extend 8 bits to 32 and add
UXTAB16	Dual extend 8 bits to 16 and add
UXTAH	Extend 16 bits to 32 and add
UXTB	Zero extend a byte
UXTB16	Dual zero extend 8 bits to 16 and add
UXTH	Zero extend a halfword

12.9.1.9 System Handler Priority Register 1

Name: SCB_SHPR1

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
PRI_6							

15	14	13	12	11	10	9	8
PRI_5							

7	6	5	4	3	2	1	0
PRI_4							

- **PRI_6: Priority**

Priority of system handler 6, UsageFault.

- **PRI_5: Priority**

Priority of system handler 5, BusFault.

- **PRI_4: Priority**

Priority of system handler 4, MemManage.

12.11.2.2 MPU Control Register

Name: MPU_CTRL

Access: Read/Write

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

The MPU CTRL register enables the MPU, enables the default memory map background region, and enables the use of the MPU when in the hard fault, Non-maskable Interrupt (NMI), and FAULTMASK escalated handlers.

- **PRIVDEFENA: Privileged Default Memory Map Enable**

Enables privileged software access to the default memory map:

0: If the MPU is enabled, disables the use of the default memory map. Any memory access to a location not covered by any enabled region causes a fault.

1: If the MPU is enabled, enables the use of the default memory map as a background region for privileged software accesses.

When enabled, the background region acts as a region number -1. Any region that is defined and enabled has priority over this default map.

If the MPU is disabled, the processor ignores this bit.

- **HFNMIENA: Hard Fault and NMI Enable**

Enables the operation of MPU during hard fault, NMI, and FAULTMASK handlers.

When the MPU is enabled:

0: MPU is disabled during hard fault, NMI, and FAULTMASK handlers, regardless of the value of the ENABLE bit.

1: The MPU is enabled during hard fault, NMI, and FAULTMASK handlers.

When the MPU is disabled, if this bit is set to 1, the behavior is unpredictable.

- **ENABLE: MPU Enable**

Enables the MPU:

0: MPU disabled.

1: MPU enabled.

When ENABLE and PRIVDEFENA are both set to 1:

- For privileged accesses, the *default memory map* is as described in “Memory Model”. Any access by privileged software that does not address an enabled memory region behaves as defined by the default memory map.
- Any access by unprivileged software that does not address an enabled memory region causes a memory management fault.

12.11.2.3 MPU Region Number Register

Name: MPU_RNR

Access: Read/Write

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

The MPU_RNR selects which memory region is referenced by the MPU_RBAR and MPU_RASRs.

- **REGION: MPU Region Referenced by the MPU_RBAR and MPU_RASRs**

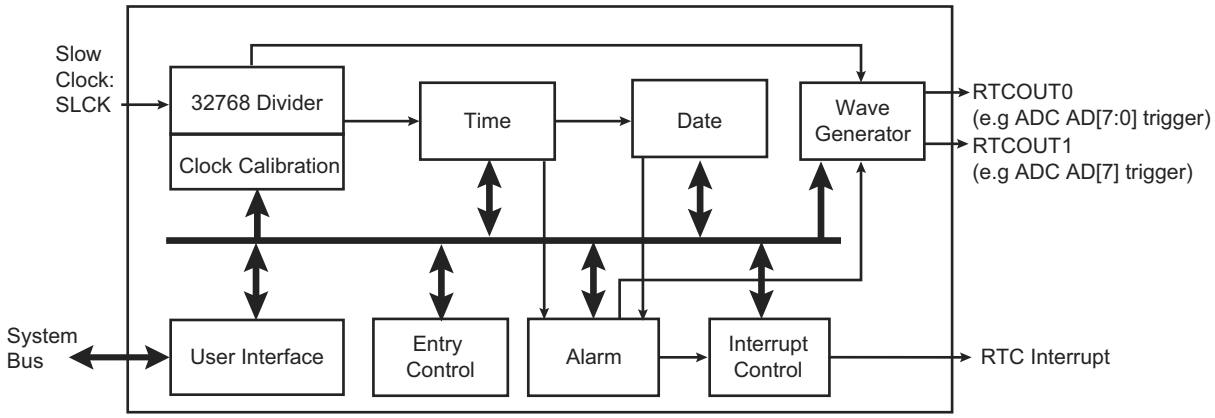
Indicates the MPU region referenced by the MPU_RBAR and MPU_RASRs.

The MPU supports 8 memory regions, so the permitted values of this field are 0–7.

Normally, the required region number is written to this register before accessing the MPU_RBAR or MPU_RASR. However, the region number can be changed by writing to the MPU_RBAR with the VALID bit set to 1; see “MPU Region Base Address Register”. This write updates the value of the REGION field.

16.3 Block Diagram

Figure 16-1. Real-time Clock Block Diagram



16.4 Product Dependencies

16.4.1 Power Management

The Real-time Clock is continuously clocked at 32.768 kHz. The Power Management Controller has no effect on RTC behavior.

16.4.2 Interrupt

RTC interrupt line is connected on one of the internal sources of the interrupt controller. RTC interrupt requires the interrupt controller to be programmed first.

Table 16-1. Peripheral IDs

Instance	ID
RTC	2

16.5 Functional Description

The RTC provides a full binary-coded decimal (BCD) clock that includes century (19/20), year (with leap years), month, date, day, hours, minutes and seconds reported in RTC Time Register (RTC_TIMR) and RTC Calendar Register (RTC_CALR).

The valid year range is up to 2099 in Gregorian mode (or 1300 to 1499 in Persian mode).

The RTC can operate in 24-hour mode or in 12-hour mode with an AM/PM indicator.

Corrections for leap years are included (all years divisible by 4 being leap years except 1900). This is correct up to the year 2099.

The RTC can generate events to trigger ADC measurements.

16.5.1 Reference Clock

The reference clock is the Slow Clock (SLCK). It can be driven internally or by an external 32.768 kHz crystal.

During low power modes of the processor, the oscillator runs and power consumption is critical. The crystal selection has to take into account the current consumption for power saving and the frequency drift due to temperature effect on the circuit for time accuracy.

- **TDERR: Time and/or Date Free Running Error**

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

28.7 Functional Description

28.7.1 Modes of Operation

The SPI operates in Master mode or in Slave mode.

- The SPI operates in Master mode by writing a 1 to the MSTR bit in the SPI Mode Register (SPI_MR):
  - Pins NPCS0 to NPCS1 are all configured as outputs
  - The SPCK pin is driven
  - The MISO line is wired on the receiver input
  - The MOSI line is driven as an output by the transmitter.
- The SPI operates in Slave mode if the MSTR bit in the SPI_MR is written to 0:
  - The MISO line is driven by the transmitter output
  - The MOSI line is wired on the receiver input
  - The SPCK pin is driven by the transmitter to synchronize the receiver.
  - The NPCS0 pin becomes an input, and is used as a slave select signal (NSS)
  - Pin NPCS1 is not driven and can be used for other purposes.

The data transfers are identically programmable for both modes of operation. The baud rate generator is activated only in Master mode.

28.7.2 Data Transfer

Four combinations of polarity and phase are available for data transfers. The clock polarity is programmed with the CPOL bit in the SPI Chip Select register (SPI_CSR). The clock phase is programmed with the NCPHA bit. These two parameters determine the edges of the clock signal on which data is driven and sampled. Each of the two parameters has two possible states, resulting in four possible combinations that are incompatible with one another. Consequently, a master/slave pair must use the same parameter pair values to communicate. If multiple slaves are connected and require different configurations, the master must reconfigure itself each time it needs to communicate with a different slave.

Table 28-4 shows the four modes and corresponding parameter settings.

Table 28-4. SPI Bus Protocol Modes

SPI Mode	CPOL	NCPHA	Shift SPCK Edge	Capture SPCK Edge	SPCK Inactive Level
0	0	1	Falling	Rising	Low
1	0	0	Rising	Falling	Low
2	1	1	Rising	Falling	High
3	1	0	Falling	Rising	High

Figure 28-3 and Figure 28-4 show examples of data transfers.

29.7.2 TWIHS Master Mode Register

Name: TWIHS_MMR

Address: 0x40018004

Access: Read/Write

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	DADR						

15	14	13	12	11	10	9	8
–	–	–	MREAD	–	–	IADRSZ	

7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–

- **IADRSZ: Internal Device Address Size**

Value	Name	Description
0	NONE	No internal device address
1	1_BYTE	One-byte internal device address
2	2_BYTE	Two-byte internal device address
3	3_BYTE	Three-byte internal device address

- **MREAD: Master Read Direction**

0: Master write direction.

1: Master read direction.

- **DADR: Device Address**

The device address is used to access slave devices in Read or Write mode. These bits are only used in Master mode.

29.7.13 TWIHS Transmit Holding Register

Name: TWIHS_THR

Address: 0x40018034

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
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
TXDATA							

- **TXDATA: Master or Slave Transmit Holding Data**

30.7.3 Master Mode

30.7.3.1 Definition

The master is the device that starts a transfer, generates a clock and stops it.

30.7.3.2 Programming Master Mode

The following fields must be programmed before entering Master mode:

1. TWI_MMR.DADR (+ IADRSZ + IADR if a 10-bit device is addressed): The device address is used to access slave devices in Read or Write mode.
2. TWI_CWGR.CKDIV + CHDIV + CLDIV: Clock waveform.
3. TWI_CR.SVDIS: Disables the Slave mode
4. TWI_CR.MSEN: Enables the Master mode

Note: If the TWI is already in Master mode, the device address (DADR) can be configured without disabling the Master mode.

30.7.3.3 Master Transmitter Mode

After the master initiates a START condition when writing into the Transmit Holding register (TWI_THR), it sends a 7-bit slave address, configured in the Master Mode register (DADR in TWI_MMR), to notify the slave device. The bit following the slave address indicates the transfer direction—0 in this case (MREAD = 0 in TWI_MMR).

The TWI transfers require the slave to acknowledge each received byte. During the acknowledge clock pulse (9th pulse), the master releases the data line (HIGH), enabling the slave to pull it down in order to generate the acknowledge. If the slave does not acknowledge the byte, then the Not Acknowledge flag (NACK) is set in the TWI Status Register (TWI_SR) of the master and a STOP condition is sent. The NACK flag must be cleared by reading the TWI Status Register (TWI_SR) before the next write into the TWI Transmit Holding Register (TWI_THR). As with the other status bits, an interrupt can be generated if enabled in the Interrupt Enable register (TWI_IER). If the slave acknowledges the byte, the data written in the TWI_THR is then shifted in the internal shifter and transferred. When an acknowledge is detected, the TXRDY bit is set until a new write in the TWI_THR.

TXRDY is used as Transmit Ready for the PDC transmit channel.

When no more data is written into the TWI_THR, the master generates a STOP condition to end the transfer. A TXCOMP bit value of one in the TWI_SR indicates that the transfer has completed. See Figure 30-4, Figure 30-5, and Figure 30-6.

To clear the TXRDY flag, first set the bit TWI_CR.MSDIS, then set the bit TWI_CR.MSEN.

Figure 30-4. Master Write with One Data Byte

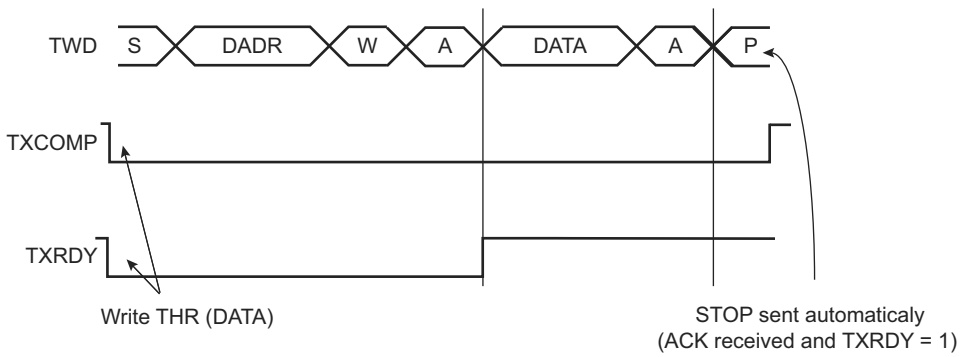


Figure 31-6. Dual Microphone Application Block Diagram

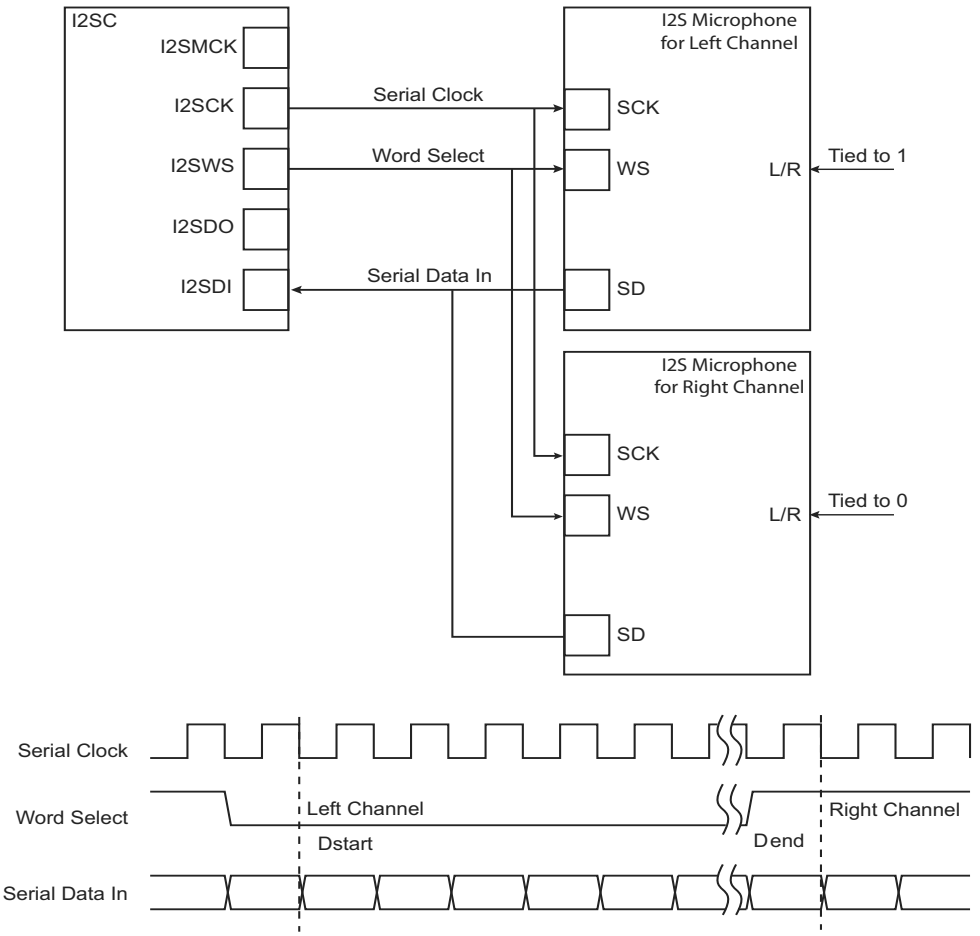
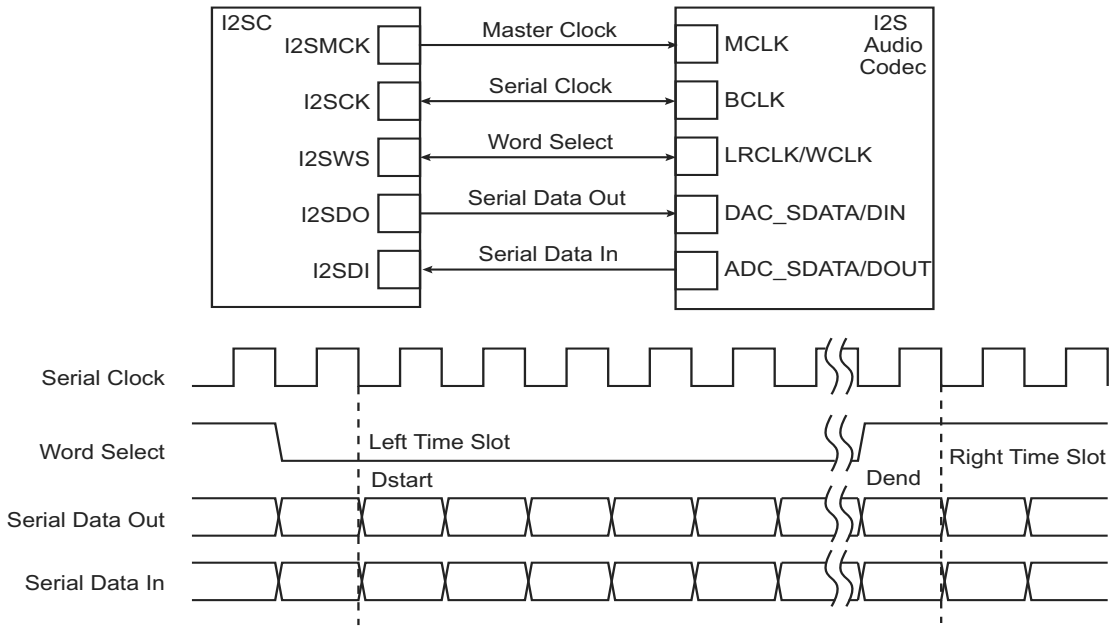


Figure 31-7. Codec Application Block Diagram



33.7.1 USART Control Register

Name: US_CR

Address: 0x40024000

Access: Write-only

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	RTSDIS	RTSEN	–	–

15	14	13	12	11	10	9	8
RETTO	–	–	SENDA	STTTO	STPBRK	STTBRK	RSTSTA

7	6	5	4	3	2	1	0
TXDIS	TXEN	RXDIS	RXEN	RSTTX	RSTRX	–	–

For SPI control, see Section 33.7.2 ”USART Control Register (SPI_MODE)”.

- **RSTRX: Reset Receiver**

0: No effect.

1: Resets the receiver.

- **RSTTX: Reset Transmitter**

0: No effect.

1: Resets the transmitter.

- **RXEN: Receiver Enable**

0: No effect.

1: Enables the receiver, if RXDIS is 0.

- **RXDIS: Receiver Disable**

0: No effect.

1: Disables the receiver.

- **TXEN: Transmitter Enable**

0: No effect.

1: Enables the transmitter if TXDIS is 0.

- **TXDIS: Transmitter Disable**

0: No effect.

1: Disables the transmitter.

- **RSTSTA: Reset Status Bits**

0: No effect.

1: Resets the status bits PARE, FRAME, OVRE and RXBRK in US_CSR.

CPOL is used to determine the inactive state value of the serial clock (SPCK). It is used with CPHA to produce the required clock/data relationship between master and slave devices.

- **CLKO: Clock Output Select**

0: The USART does not drive the SCK pin.

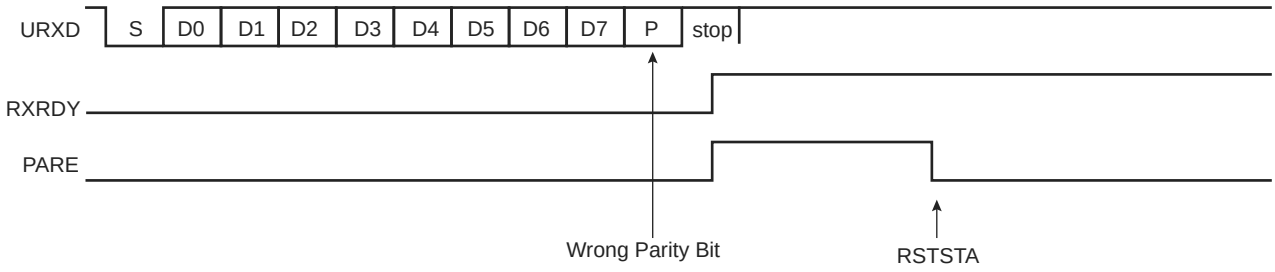
1: The USART drives the SCK pin if USCLKS does not select the external clock SCK.

- **WRDBT: Wait Read Data Before Transfer**

0: The character transmission starts as soon as a character is written into US_THR (assuming TXRDY was set).

1: The character transmission starts when a character is written and only if RXRDY flag is cleared (Receive Holding Register has been read).

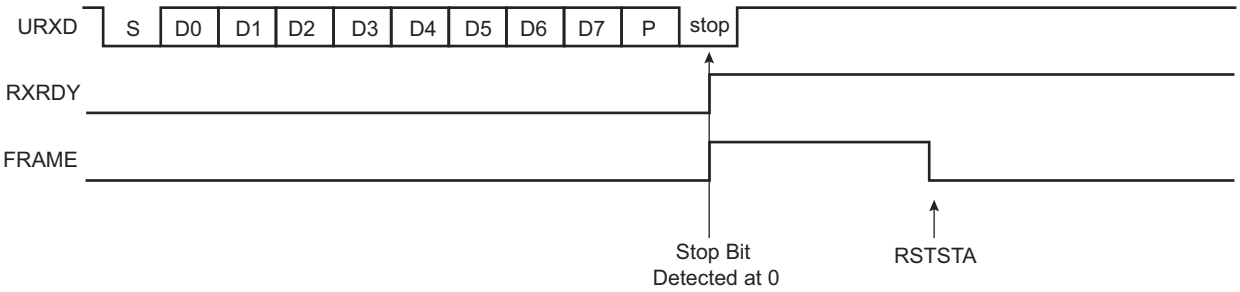
Figure 34-7. Parity Error



34.5.2.6 Receiver Framing Error

When a start bit is detected, it generates a character reception when all the data bits have been sampled. The stop bit is also sampled and when it is detected at 0, the FRAME (Framing Error) bit in UART_SR is set at the same time the RXRDY bit is set. The FRAME bit remains high until the Control Register (UART_CR) is written with the bit RSTSTA at 1.

Figure 34-8. Receiver Framing Error



34.5.2.7 Receiver Digital Filter

The UART embeds a digital filter on the receive line. It is disabled by default and can be enabled by writing a logical 1 in the FILTER bit of UART_MR. When enabled, the receive line is sampled using the 16x bit clock and a three-sample filter (majority 2 over 3) determines the value of the line.

34.5.3 Transmitter

34.5.3.1 Transmitter Reset, Enable and Disable

After device reset, the UART transmitter is disabled and must be enabled before being used. The transmitter is enabled by writing UART_CR with the bit TXEN at 1. From this command, the transmitter waits for a character to be written in the Transmit Holding Register (UART_THR) before actually starting the transmission.

The programmer can disable the transmitter by writing UART_CR with the bit TXDIS at 1. If the transmitter is not operating, it is immediately stopped. However, if a character is being processed into the internal shift register and/or a character has been written in the UART_THR, the characters are completed before the transmitter is actually stopped.

The programmer can also put the transmitter in its reset state by writing the UART_CR with the bit RSTTX at 1. This immediately stops the transmitter, whether or not it is processing characters.

34.5.3.2 Transmit Format

The UART transmitter drives the pin UTXD at the baud rate clock speed. The line is driven depending on the format defined in UART_MR and the data stored in the internal shift register. One start bit at level 0, then the 8 data bits, from the lowest to the highest bit, one optional parity bit and one stop bit at 1 are consecutively shifted

36. Analog-to-Digital Converter (ADC)

36.1 Description

The ADC is based on a 10-bit Analog-to-Digital Converter (ADC) managed by an ADC Controller providing enhanced resolution up to 12 bits. Refer to Figure 36-1 “Analog-to-Digital Converter Block Diagram”. It also integrates a 8-to-1 analog multiplexer, making possible the analog-to-digital conversions of 8 analog lines. The conversions extend from 0V to VDDIO.

The ADC digital controller embeds circuitry to reduce the resolution down to 8 bits. The 8-bit resolution mode prevents using 16-bit Peripheral DMA transfer into memory when only 8-bit resolution is required by the application. Note that using this low resolution mode does not increase the conversion rate.

Conversion results are reported in a common register for all channels, as well as in a channel-dedicated register.

The 11-bit and 12-bit resolution modes are obtained by averaging multiple samples to decrease quantization noise. For the 11-bit mode, 4 samples are used, which gives a real sample rate of 1/4 of the actual sample frequency. For the 12-bit mode, 16 samples are used, giving a real sample rate of 1/16 of the actual sample frequency. This arrangement allows conversion speed to be traded for better accuracy.

Software trigger, external trigger on rising edge of the ADTRG pin or internal triggers from Timer Counter output(s) are configurable.

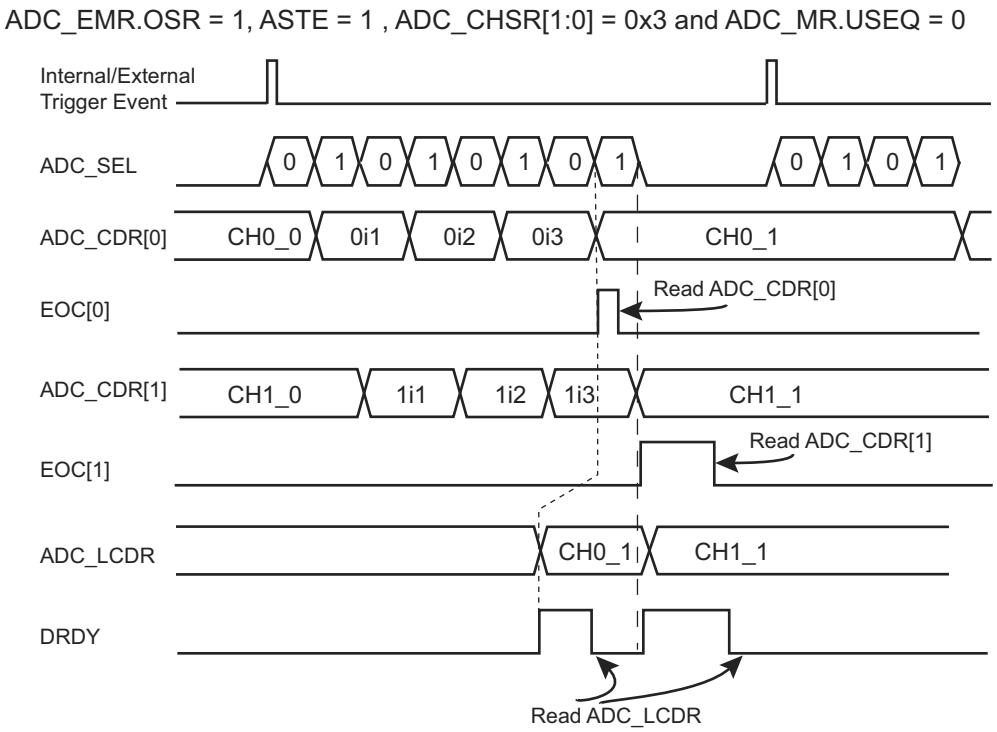
The last channel can be converted at a rate different from other channels to improve conversion and processing efficiency in case of a device which provides very low frequency variations such as a temperature sensor. A dedicated comparison circuitry on the last channel allows specific processing and interrupt.

The main comparison circuitry allows automatic detection of values below a threshold, higher than a threshold, in a given range or outside the range, thresholds and ranges being fully configurable.

The ADC also integrates a Sleep mode and a conversion sequencer and connects with a PDC channel. These features reduce both power consumption and processor intervention.

Finally, the user can configure ADC timings, such as startup time and tracking time.

Figure 36-10. Digital Averaging Function Waveforms on a Single Trigger Event



Note: ADC_SEL: Command to the ADC analog cell
0i1, 0i2, 0i3, 1i1, 1i2, 1i3 are intermediate results and CH0/1_0/1 are final result of average function.

When USEQ is set, the user can define the channel sequence to be converted by configuring ADC_SEQRx and ADC_CHER so that channels are not interleaved during the averaging period. Under these conditions, a sample is defined for each end of conversion as described in Figure 36-11.

Therefore, if the same channel is configured to be converted four times consecutively, and the OSR = 1 in ADC_EMR, the averaging result will be placed in the corresponding ADC_CDRx and in ADC_LCDR for each trigger event.

In this case, the ADC real sample rate remains the maximum ADC sample rate divided by 4 or 16, depending on OSR.

When USEQ = 1, ASTE = 1 and OSR is different from 0, it is important to notice that the user sequence must follow a specific pattern. The user sequence must be programmed in such a way that it generates a stream of conversion, where a given channel is successively converted with an integer multiple depending on OSR value. Up to four channels can be converted in this specific mode.

When OSR = 1, each channel to convert must be consecutively repeated four times in the sequence, so the four first single bit field enabled in ADC_CHSR must have their associated channel index programmed to the same value in ADC_SEQRx registers. Therefore, for OSR = 1 a maximum of four channels can be converted (the user sequence allows a maximum of 16 conversions for each trigger event).

When OSR = 2, a channel to convert must be consecutively repeated 16 times in the sequence, so all fields must be enabled in ADC_CHSR, and their associated channel index programmed to the same value in the ADC_SEQRx registers. Therefore, for OSR = 2 only one channel can be converted (the user sequence allows a maximum of 16 conversions for each trigger event).

OSR = 3 and OSR = 4 are prohibited when USEQ = 1 and ASTE = 1.

Table 42-2. SAM G53 Datasheet Rev. 11240E Revision History (Continued)

Doc. Rev. 11240E	Changes
17-Oct-14	Section 14. “Real-time Timer (RTT)”
	Section 15.4 “Functional Description”: replaced "... the CRTV field must be read twice at the same value to improve accuracy of the returned value." with "...the CRTV field must be read twice at the same value to read a correct value."
	Section 15.5.3 “Real-time Timer Value Register”: added note.
	Section 15.5.4 “Real-time Timer Status Register”: added “(cleared on read)” to each bit description
	Section 17. “Real-time Clock (RTC)”
	Section 17.5.5 “RTC Internal Free Running Counter Error Checking”: replaced “RTC status clear control register” with “Status Clear Command Register”
	Section 20. “Supply Controller (SUPC)”
	Section 20.4.2 ”Slow Clock Generator”: added “prior to writing a 1 in bit XTALSEL” to “In order to set the bypass mode, the OSCBYPASS bit of the SUPC Mode Register (SUPC_MR) needs to be set to 1.”
	Section 20.5.5 ”Supply Controller Mode Register”: modified OSCBYPASS bit description
	Section 20.5.6 ”Supply Controller Status Register”: added “cleared on read” to relevant bit descriptions
	Section 22. “General Purpose Backup Registers (GPBR)”
	Section 22.3.1 “General Purpose Backup Register x”: "VDDIO" replaced with "VDDBU" in "These registers are reset at first power-up and on each loss of VDDBU"
	Section 26. ”Enhanced Embedded Flash Controller (EEFC)”
	Updated Figure 26-2 “Code Read Optimization for FWS = 0” and Figure 26-3 “Code Read Optimization for FWS = 3”
	Figure 26-6 “Command State Chart” : replaced two instances of “MC_FSR” with “EEFC_FSR”
	Section 26.4.3.2 “Write Commands”: replaced “When the Programming Page command is given” with “When a ‘Write Page’ (WP) command is issued”
	Section 26.4.3.3 “Erase Commands”: replaced “When programming is completed” with “When erasing is completed”; replaced “after a programming sequence” with “after an erasing sequence”; replaced “Flash Error: At the end of the programming” with “Flash Error: At the end of the erase period”; replaced “modulo” with “a multiple of”
	Section 26.4.3.8 “Unique Identifier”: added “(see table “Unique Identifier Definition” in section “Memories”)” to end of first sentence. Added “Field EEFC_FCR.FARG is meaningless” to end of first and fourth steps of read sequence. Added sentence “The ‘Start Read Unique Identifier’ command reuses some addresses of the memory plane but the Unique Identifier is physically different from the memory plane” to third step of unique identifier read sequence
	Section 26.4.3.9 “User Signature”: added “Field EEFC_FCR.FARG is meaningless” to end of first and fourth steps of read sequence, to end of second step of write sequence, and to end of first step of erase sequence. Added sentence “The ‘Start Read User Signature’ command reuses some addresses of the memory plane but the User Signature is physically different from the memory plane.” to third step of user signature read sequence
	Section 26.5.1 “EEFC Flash Mode Register”: replaced “No Flash read should be done during change of this register” with “No Flash read should be done during change of this field”
	Section 26.5.2 “EEFC Flash Command Register”: corrected FARG description for EPA command
	Section 26.5.3 “EEFC Flash Status Register”: added “(cleared...)” to each bit description
	Section 28. “Bus Matrix (MATRIX)”
	Table 28-4 “Register Mapping”: corrected offset values. Added one note to address offset 0x0110