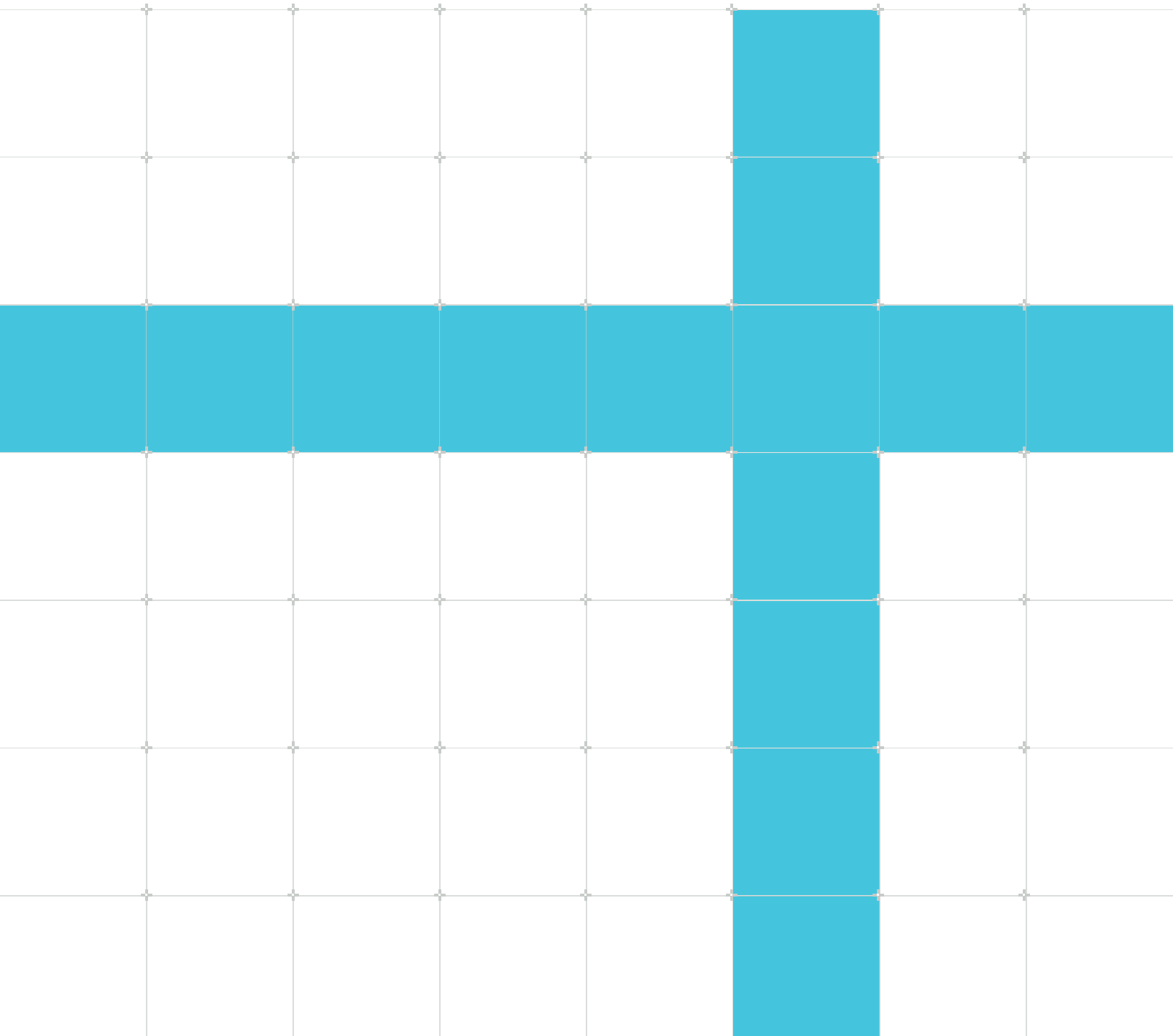
Arm® Security Alarm Manager

Version 1.0

Specification

Issue 01
107716_0000_01_en



Arm® Security Alarm Manager

Specification

Release information

Document history

Issue	Date	Confidentiality	Change
0000-01	31 May 2023	Non-Confidential	Initial release

Proprietary Notice

and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word "partner" in reference to Arm's customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349|version 21.0)

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Feedback

Arm welcomes feedback on this product and its documentation. To provide feedback on the product, create a ticket on https://support.developer.arm.com

To provide feedback on the document, fill the following survey: https://developer.arm.com/documentation-feedback-survey.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

We believe that this document contains no offensive language. To report offensive language in this document, email terms@arm.com.

Contents

1. Introduction

1.1 Conventions

The following subsections describe conventions used in Arm documents.

Glossary

The Arm Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: developer.arm.com/glossary.

Typographic conventions

Arm documentation uses typographical conventions to convey specific meaning.

Convention	Use
italic	Citations.
bold	Interface elements, such as menu names. Terms in descriptive lists, where appropriate.
`monospace`	Text that you can enter at the keyboard, such as commands, file and program names, and source code.
`monospace underline`	A permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
`<and>`	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2>`
SMALL CAPITALS	Terms that have specific technical meanings as defined in the *Arm® Glossary*. For example, **IMPLEMENTATION DEFINED**, **IMPLEMENTATION SPECIFIC**, **UNKNOWN**, and **UNPREDICTABLE**.
	Recommendations. Not following these recommendations might lead to system failure or damage.
	Requirements for the system. Not following these requirements might result in system failure or damage.
	Requirements for the system. Not following these requirements will result in system failure or damage.
	An important piece of information that needs your attention.

Convention	Use
	A useful tip that might make it easier, better or faster to perform a task.
	A reminder of something important that relates to the information you are reading.

1.2 Useful resources

This document contains information that is specific to this product. See the following resources for other useful information.

Access to Arm documents depends on their confidentiality:

- Non-Confidential documents are available at developer.arm.com/documentation. Each document link in the following tables goes to the online version of the document.
- Confidential documents are available to licensees only through the product package.

Arm product resources	Document ID	Confidentiality
Arm® CoreLink™ DMA-350 Controller Technical Reference Manual	102482	Non-Confidential
Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual	DDI 0571	Non-Confidential

Arm architecture and specifications	Document ID	Confidentiality
Arm® Lifecycle Manager Specification	107616	Non-Confidential

Arm tests its PDFs only in Adobe Acrobat and Acrobat Reader. Arm cannot guarantee the quality of its documents when used with any other PDF reader.

Adobe PDF reader products can be downloaded at http://www.adobe.com

1.3 Other information

See the Arm website for other relevant information.

- Arm® Developer.
- Arm® Documentation.
- Technical Support.
- Arm® Glossary.

2. SAM overview

The Arm® *Security Alarm Manager* (SAM) lets you apply the programmed response to the detection of up to 64 security events.

Some of these security events are digitally detected within the subsystem, and other optional events can be detected by external analog and digital sensors which are integrated by the SoC. It may take the external analog sensors time to stabilize, therefore the SoC is required to provide an External Sensors Ready signal. The External Sensors Ready signal is set once all of the external sensors are stable to avoid false alarm detection by the SAM.

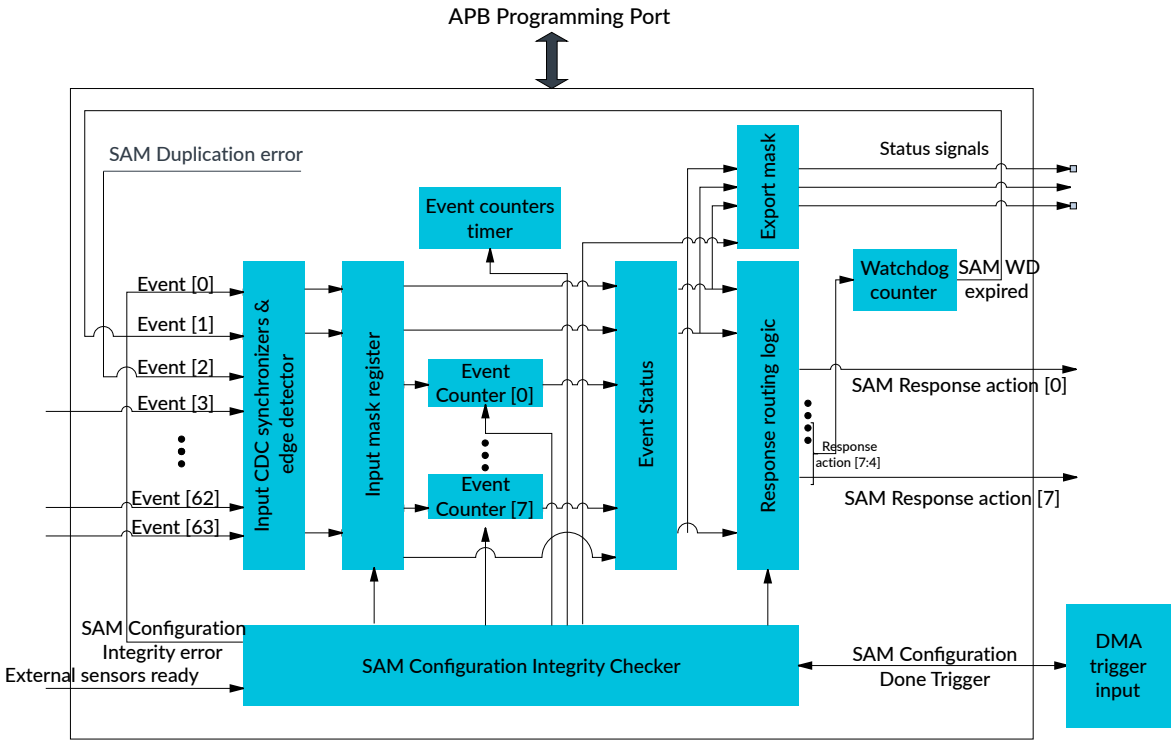
The SAM integrates platform-specific event logging registers for sources that provide many event signals. These Platform-specific event logging registers coalesce their inputs into a SAM input event.

2.1 Topology

This topics provides information about the SAM components.

Figure 2-1: SAM high-level scheme on page 10 shows a high-level scheme of the components of the SAM.

Figure 2-1: SAM high-level scheme



The hardware components included in the SAM are:

Configuration integrity checker

Performs an integrity check for the shadow registers in the ICV protected register block. If the integrity check is successful, the configuration integrity checker forwards the values of these registers to the respective functional registers of the SAM, otherwise it would signal an alarm.

Configuration done trigger unit

Asserts the trigger when the SAM configuration has successfully completed loading after exit from Cold reset.

Input masking

Allows for a programmatic way to mask out unstable or unconnected event inputs.

Export masking

Exports the unmasked event status bits to allow the SoC to understand the subsystem's status while not revealing masked out details.

Input clock domain crossing synchronizers and edge detectors

The SAM implements *clock domain crossing* (CDC) synchronization for all the event input signals to handle external events that are not synchronized with the SAM clock.

Response routing logic

Allows the SAM to uniquely route each of the 64 event statuses, which are latched in the SAMES0 and SAMES1 registers, to one of the available response action output signals.

Integrated watchdog counter

Asserts a dedicated watchdog interrupt when response actions 2 to 7 of the SAM response action allocation are not cleared before the watchdog counter expires. Response actions 2 to 7 are typically connected to the trusted processor interrupt lines of the system, either as an *Non-Maskable Interrupt* (NMI) or as a normal interrupt. This signal is connected inside the SAM as event 1 which can be routed to a reset response. The Watchdog interrupt signal is connected inside the SAM as event 1 which can be routed to a reset response.

Event counters

Monitor the incoming input events and assert the respective event status. A respective event status can only be asserted when the number of input events connected to the counter has been accumulated and the configured threshold of the counter has been obtained. The SAM supports four event counters where each counter can be assigned an input event.

Event counters timer

The event counters timer paces the time of the event counters' power-on values that are loaded by the subsystem to the SAM event counter registers.

2.2 SAM operation

The input events pass through clock domain crossing synchronizer and edge detector. The SAM monitors the incoming enabled events and sets the corresponding bit in the SAM events status

registers when the corresponding edge is detected. After the security event is serviced, the software clears the captured event.

The flow of the SAM operation is as follows:

1. Incoming events pass through the clock domain crossing synchronizer and edge detector.
2. The SAM monitors the incoming enabled events according to the setup of SAMIM0 and SAMIM1 registers. When an edge is detected it sets the corresponding bit in the SAM events status registers (SAMES0 and SAMES1).
3. For each response, if one of the status bits in the SAMES0 and SAMES1 registers is set, then a response output signal is asserted. The response output signal is configured by its respective event response action routing bit in SAM response routing logic register (SAMRRLSn (n=0..7)).
4. The indicated events in the SAM events status registers (SAMES0 and SAMES1) pass through an export mask, configured by the SAMEM0 and SAMEM1 registers, to the output event status interface signals. The output event status interface signals are visible to the SoC. For security reasons, only those alerts, which are not masked, are indicated as out. The SoC integrator can accumulate some of the exported statuses to fewer status signals, and can present them as a set of *Persistent State Information* (PSI) signals for its needs.
5. When the security event is serviced, the software clears the input event at its source, unless the event source generates a pulse and does not need clearing. For example a bus error or an uncorrectable memory error do not need clearing, as their response can be reset which clears their event.
6. The runtime software clears the captured event by setting the corresponding clear bit in the SAM event clear registers (SAMECL0 and SAMECL1). Setting the corresponding clear bit concurrently clears the event bit in SAM events status registers, and causes the clock domain crossing synchronizer to reevaluate the respective input event.
7. Reevaluating the input event verifies that the input was set after it was serviced and before the software cleared its bit in SAM events status registers. The software clears its bit in SAM events status registers by setting its bit in the SAM event clear registers, to prevent loss of a new event.

2.3 ICV protected registers

The SAM control and status registers are programmed through its AMBA APB3 compliant subordinate interface.

The SAM registers are divided into two types of register sets:

- Regular registers
- *Integrity Checked Value* (ICV) protected registers

Each ICV protected register has a shadow register which does not affect the functional register. The shadow registers are copied into the functional registers only if the integrity check of their set meets the value in the SAMICV shadow register.

When the reset of the SAM, COLDRESETn, is released, the hardware resets the default values of the SAM registers which provide protection for the configuration of the SAM.

At boot after release from reset, the boot process must set up the SAM. This typical boot process is as follows:

1. Read the configuration from an NVM.
2. Write the values into the respective SAM ICV registers. The last register read from the configuration and written to the SAM is to the SAMICV register.

   The written values are initially held in the shadow registers. Shadow registers are temporary, not visible registers that are used by the SAM as temporary storage until their content integrity is validated.
3. The Configuration integrity checker is triggered to compute the integrity check value over the shadow registers and compare the result to the value in SAMICV.
   - If the values match, the shadow registers are copied to the functional registers.
   - If the values do not match, the SAM registers the attack and performs a Cold reset on the system. The shadow registers are not copied to the functional registers.

Reads and writes of SAM registers in the boot process are often performed by a trusted processor in the system. However, to achieve higher security, an engine, such as a DMA device, can be used to perform reads and writes, while the processor is held from booting after reset. The SAM Configuration Done signal can then signal to the DMA device to allow the processor to boot.

To protect SAM against attacks, its register values are duplicated to inverse polarity registers so that an inverse value is written to the duplicate register. The duplicated register outputs are always compared so that they match each other's inverse value. A detected mismatch on any of the SAM registers would set sam_duperr alarm signal.

For more information about the sam_duperr alarm signal, see 5. SAM incoming events allocation on page 28.

2.4 Platform-specific event logging registers

The SAM integrates platform-specific event logging registers for sources that can provide one of more event signals set by the source, for example the processor or the DMA. These platform-specific event logging registers coalesce their inputs into a few SAM input events.

The platform-specific event logging registers are:

- SAMCDRES
- SAMRRES0
- SAMRRES1
- SAMRRES2
- VMPWCA<n>, Volatile Memory Partial Write Captured Address register
- VMSCEECA<n>, Volatile Memory Single Corrected ECC Error Captured Address register
- VMDUEECA<n>, Volatile Memory Double Uncorrected ECC Error Captured Address register
- TRAMSCEECA, TRAM Single Corrected ECC Error Captured Address register

- TRAMDUEECA, TRAM Double Uncorrected ECC Error Captured Address register

3. Functional description

The SAM has several components that provide different functionality.

3.1 Configuration integrity checker

The configuration integrity checker performs an integrity check on the shadow registers in the ICV protected register block. If the integrity check is successful, this checker forwards their values to the respective functional registers of the SAM.

The workflow of the SAM configuration integrity checker:

1. The SAM configuration integrity checker is triggered to perform its operation every time the SAMICV shadow register is written.
    - If the integrity check succeeds, the SAM:
        a. Waits for the external sensors to stabilize, which is indicated by the external sensors ready signal (SAMSENSORREADY) being set.
        b. Copies the shadow registers to the functional registers
        c. Sets the SAM configuration done trigger to signal that the boot process can continue.
    - If the integrity check is corrupted, the SAM does not set the configuration done trigger signals, to avoid signalling the continuation of the boot process. Instead, it sets the configuration integrity error signal which sets input Event[0] of the SAM. This error signal causes the SAM to:
        - Subject to the appropriate hardware defaults, the SAM propagates the event to SAMES0[0] register bit.
        - Subject to the appropriate Response Routing Logic defaults, the SAM sets Response Action zero (SAMCRSTREQ) to the Cold reset logic of the subsystem. The Response Action zero is connected at the system level.

The workflow of the SAM configuration integrity checker for an unprovisioned SoC:

1. In an unprovisioned SoC, the SAM shadow registers values are typically loaded with zeros since the configuration data is read from an uninitialized NVM.
2. When the SAM configuration integrity checker finds that all the values it checks are zeros, including the integrity check value in SAMICV, it realizes that it is an unprovisioned system.
3. The configuration integrity checker sets an internal All Zeros flag.
    - If the All Zeros flag is set, it:
        - Prevents the transition of the zero filled shadow registers to the functional registers
        - Lets the hardware defaults remain
    - If the All Zeros flag is not set and the integrity check is successful, the shadow registers of the registers that are not locked are copied into their functional registers. The registers which are not locked are defined by the values in the functional SAMRL register.


Note

This operation does not use the value in the shadow SAMRL register.

4. When the external sensors ready signal is set, it uses the SAM configuration done signals sequence to allow the boot process to continue.

If there is no reset or another trigger to restart the operation of the SAM configuration integrity checker, it does not:

- Check the external sensors ready signal again
- Initiate a new trigger sequence at its configuration done trigger signals

The following table lists the SAM configuration integrity checker shadow registers loaded from the NVM.

Table 3-1: SAM Configuration Integrity Check register value offsets in the SAM section of the NVM

NVM Offset in SAM section	Entry name (Target SAM register / use)	Size	Comment
0x00	SAMEM0	32 bits	
0x04	SAMEM1	32 bits	
0x08	SAMIM0	32 bits	
0x0C	SAMIM1	32 bits	
0x10	SAMRRLS0	32 bits	
0x14	SAMRRLS1	32 bits	
0x18	SAMRRLS2	32 bits	
0x1C	SAMRRLS3	32 bits	
0x20	SAMRRLS4	32 bits	
0x24	SAMRRLS5	32 bits	
0x28	SAMRRLS6	32 bits	
0x2C	SAMRRLS7	32 bits	
0x30	SAMEC0	32 bits	

NVM Offset in SAM section	Entry name (Target SAM register / use)	Size	Comment
0x34	SAMEC1	32 bits	If the SAMNEC is < 1, this field is still required. Any value may be used.
0x38	SAMEC2	32 bits	If the SAMNEC is < 2, this field is still required. Any value may be used.
0x3C	SAMEC3	32 bits	If the SAMNEC is < 3, this field is still required. Any value may be used.
0x40	SAMEC4	32 bits	If the SAMNEC is < 4, this field is still required. Any value may be used.
0x44	SAMEC5	32 bits	If the SAMNEC is < 5, this field is still required. Any value may be used.
0x48	SAMEC6	32 bits	If the SAMNEC is < 6, this field is still required. Any value may be used.
0x4C	SAMEC7	32 bits	If the SAMNEC is < 7, this field is still required. Any value may be used.
0x50	SAMECTIV	32 bits	
0x54	SAMWDCIV	32 bits	
0x58	SAMRL	32 bits	
0x5C	SAMICV	32 bits	Integrity check value for the SAM shadow registers. It is a zero-count value of all shadow registers. There are 736 bits which are protected by the SAMICV, therefore 10 bits from this 32-bit register are used for the zero count.

3.2 Configuration done trigger unit

The SAM includes an configuration done trigger unit that asserts the trigger when the SAM configuration has completed loading successfully. The trigger signals are designed to work with a DMA, for example the Arm® CoreLink™ DMA-350 Controller input trigger interface.

For more information, see the trigger interface section of the *Arm® CoreLink™ DMA-350 Controller Technical Reference Manual*.

The following table lists the configuration done trigger interface signals:

Table 3-2: Configuration done trigger signals

Signal name	Description
sam_config_done_trig_req	Output from the SAM to the external trigger interface. It is asserted by the SAM configuration integrity checker when it has completed successfully.
sam_config_done_trig_ack	Input to the SAM from the external trigger interface. It is used for deasserting the trigger request.

When using this interface with the DMA-350 trigger interface, all trig_in_<TI>_req_type[1:0] and all trig_in_<TI>_ack_type[1:0] signals are tied to 0, which indicates the request type SINGLE (2'b00)

and the ack type OKAY (2'b00). This trigger is only expected to be used to trigger the start of a DMA command.

Related information

- Configuration done trigger interface

3.3 Input masking

Input masking programmatically masks out unstable or unconnected event inputs.

The input masking block uses the values in the SAMIM0 and SAMIM1 registers.

3.4 Export masking

The SAM exports all its 64 event status bits to allow the SoC to monitor the SAM's status. The event status bits are exported as signals that may be connected either directly or coalesced to fewer signals depending on the needs of the system. However, some event indications may act as hints to an attacker on what the SAM detected. Therefore, export masking programmatically masks some events according to the bit values of the SAMEM0 and SAMEM1 registers.

3.5 Input clock domain crossing synchronizers and edge detectors

The SAM receives its external events from sensors that can be asynchronous with the SAM clock.

The SAM implements clock domain crossing synchronization for all event input signals to handle external events sensors which are asynchronous.

In addition, some sensors can provide level-type events which can cause the event counters to keep counting them, even though they are already set. To make all inputs, both leveled or pulsed, counted only one time, the SAM implements edge-detection logic. The minimum duration of the input signal must be two SAM clock cycles long.

To avoid creating a race condition when software receives an event targetting the same SAM events status registers (SAMES0 and SAMES1) which is currently being cleared, the SAM delays the setting of the particular events status register bit until after it is cleared. This behavior causes the SAM events status registers (SAMES0 and SAMES1) to be cleared by the SAM event clear registers (SAMECL0 and SAMECL1) and set again by the edge detector. The output of the edge detector to the Response Routing Logic is a pulse of one clock.

3.6 Response routing logic

The response routing logic allows you to uniquely route any of the 64 event statuses which are latched in the SAMES0 and SAMES1 registers to any of the available response action output signals.

The response routing logic routes event [8n+m], that is latched in the event status registers, to any response action (0..7) according to the respective value in the SAMRRLSn (n=0..7) register in the respective fields (m=0..7).

Multiple events (from 0 to 63) may be routed to each of the response action output signals. The response Routing Logic ORs all the routed events to create the response action.

3.7 Integrated watchdog counter

The SAM supports an *integrated watchdog counter* (IWC) that asserts a dedicated watchdog interrupt when response actions 2 to 7 are not cleared before the counter of the watchdog expires.

1. The IWC is armed and triggered when one or more of the enabled response actions 2 to 7 are asserted.
   - The response action mask is controlled by the SAMWDCIV register.
   - The OR of the six enabled response actions triggers the IWC.
2. The IWC counts down from a value loaded from the SAMWDCIV register and asserts a dedicated SAM watchdog interrupt when the counter elapses.
   - The IWC stops counting and does not assert the interrupt when its input trigger (the OR of the 6 enabled response actions) is deasserted and it is armed again with the value from the SAMWDCIV for its next input trigger.

The SAMWDCIV register value is programmable to allow flexibility. The suggested value is `0xFFFF` (64K cycles). It should provide enough time for software to clear the respective event and its associated Status register bit which eventually would deassert the related response action. The software can set the new value for the SAMWDCIV register if required. The SAM watchdog interrupt signal is expected to be connected to the system's trusted processor generic interrupt controller and is also connected internally to event[1] input of the SAM. If event[1] response routing is set up to a response action which causes a reset, then the system is reset.

The runtime software should guarantee that the corresponding interrupt service routine can clear the interrupt before the IWC counting is exhausted during normal operation.


Note

If software fails to clear the interrupt within a time shorter than the watchdog expiration time, it is recommended that the event is rooted to an action that causes a reset.

3.8 Event counters

The SAM supports event counters to monitor the incoming input events. The event counters reduce the risk associated with false-positive detection if a sensor is unstable at power up and may have one or more false alarm events. The event counters also improve the system availability at the presence of unstable security events.

The number of event counters is hardware configurable (the current implementation supports four), and they can be assigned to up to the corresponding number of input events.

Configuration of the event counters is expected to be loaded into the SAM at exit from reset along with the event counters timer value (SAMECTIV). The configuration of the event counters includes the input event allocation and the count value for each counter. For example, if it is known that a sensor generates three false events within the first 4 msec from power up, and further events should be reported to the SAMES0 and SAMES1, then a counter is allocated for the sensor with count value of 3. The event counters timer is set to number of clocks which are related to 4 msec. When the sensor signals more than 3 events within 4 msec then the indicated event (because of the alarm signal) is propagated to the SAMES0 or SAMES1 register. Otherwise, when the 4 msec pass (reaches zero), all the event counters are reset, and the event signals allocated to the counters connect again directly to the SAMES0 and SAMES1.

Each event counter watches its assigned input event defined in the setup in its SAMECn (n=0..7) register.

3.9 Event counters timer

The event counters timer paces the time of the event counters' power-on values that are loaded by the subsystem to the SAM event counter registers.

The software can set the event counters timer (through SAMECTIV) if pacing the time is needed.

Once the timer expires, it resets all the event counters, which removes the event counters from all their allocated events and reconnects the events directly to SAMES0 or SAMES1 register.

The timer is expected to be configured after reset. Configuration is implementation defined. For example, it can be configured by the boot firmware of the main processor, or the DMA.

3.10 Hardware interfaces

The SAM has several hardware interfaces, each with their own roles.

The list of hardware interfaces are as follows:

Clock interface

The SAM has the SAMCLK interface.

AMBA APB3 subordinate programming port interface

Used by the subsystem software to control the SAM operation and read its status.

Input events interface

An events interface with 61 inputs.

Status interface

A status interface with 64 outputs.

Response action interface

A response action interface with 8 outputs.

Configuration done trigger interface

The SAM has a configuration done trigger interface.

Reset interface

SAM supports two resets:

- COLDRESETn for Cold reset
- PORESETn for Power-On reset.

Events capture interface

The SAM implements capture registers for various system events.

3.10.1 AMBA APB3 subordinate programming port interface

The subsystem software uses the APB3 subordinate programming port interface to control the SAM operations and read its status.

The APB3 subordinate programming port interface does not filter transactions according to their security attributes (Secure or Non-secure) or their privilege level attributes. Therefore, we recommend controlling the SAM APB3 subordinate programming port interface using an Arm APB4 TrustZone® *Peripheral Protection Controller* (PPC) for that purpose. For more details regarding the APB4 TrustZone Peripheral Protection Controller, see the *Arm® CoreLink™ SIE-200 System IP for Embedded Technical Reference Manual*.

The memory map and registers exposed in the APB3 subordinate programming port are detailed in 7. Programmers model on page 33.

3.10.2 Input events interface

The SAM has an input events interface with 61 inputs. The integrator connects its digital and analog attack detection sensors to these inputs. For the allocation of the SAM incoming events, see 5. SAM incoming events allocation on page 28.

The sensors may provide pulse indications which must be of at least two clocks wide. Level indications are supported. They require software to acknowledge the respective sensor to clear the event.

The SAM has an external sensors ready input signal, SAMSENSORREADY. The integrator must set this signal when all the external sensors are stable after powering on the system. For more

information on the external sensors ready input signal, see 3.1 Configuration integrity checker on page 15.

3.10.3 Status interface

The SAM has a Status interface with 64 outputs.

When a signal is set, it indicates that its respective sensor reported an alarm condition. The system integrator can connect these outputs for monitoring by another entity on chip or simply as monitoring output status signals.

3.10.4 Response action interface

The SAM has a response action interface with eight outputs.

When a signal is set it triggers a response action to be taken by the system. The integrator should connect each of the response signals to trigger a response action. The SAM defines some recommended actions. For more information, see 6. SAM response action allocation on page 32.

3.10.5 Configuration done trigger interface

The SAM has a configuration done trigger interface.

When the SAM is programmed successfully after reset, and the external sensors ready indication signal is set, the signal informs an optional external entity that the SAM approves the next steps of the boot process. Optional external entity can be, for example, a DMA.

The configuration done trigger interface is designed to work with DMA trigger interface which can be optionally integrated into the system. For more information on trigger interface, see the “Trigger Interface” section of the *Arm® CoreLink™ DMA-350 Controller Technical Reference Manual*.

Related information

- Configuration done trigger unit

3.10.6 Reset interface

SAM supports COLDRESETn for Cold reset, and PORESETn for Power-On Reset.

The PORESETn resets the following registers:

- SAMES0
- SAMES1
- SAMCDRES
- SAMRRES0
- SAMRRES1

- SAMRRES2
- VPMVVCA0
- VPMVVCA1
- VPMVVCA2
- VPMVVCA3
- VMSCEECA0
- VMSCEECA1
- VMSCEECA2
- VMSCEECA3
- VMDUEECA0
- VMDUEECA1
- VMDUEECA2
- VMDUEECA3
- TRAMSCEECA
- TRAMDUEECA

We recommend connecting the COLDRESETn input of the SAM to the Cold reset logic of the system.

We recommend connecting the PORESETn input to the Power-On Reset signal of the system that is the highest in the reset hierarchy. It is only asserted when the system is powered on. When the SAM resets the system by asserting response action zero (SAMCRSTREQ) or response action one (SAMWRSTREQ), its state is reset by the Cold reset logic of the subsystem. During a reset by the Cold reset, the registers listed above maintain their values to allow the software to check the reported attack or event.

3.10.7 Events capture interface

The SAM implements capture registers for various system events. These registers get their input through the events capture interface.

The registers connected to the various system events provide input signals which are tied to their intended sources at the system. An error event connected to a SAM event accompanies each set of signals. The SAM event captures the signals in the internal capture registers. These signals and registers are customized for a specific implementation, but can be used for different purposes.

The following table lists these input signals, their error event input signals, and the SAM input trigger numbers to which the events are connected.

Table 3-3: Input signals with associated events, registers, and triggers

Input signal name	Input signal capture event	Capture register	SAM input trigger
DCLSCORECOMPRES[11:0]	CPUDCLSERR	SAMCDRES	14

Input signal name	Input signal capture event	Capture register	SAM input trigger
DCLSIWICCOMPRES[5:0]	CPUDCLSERR	SAMCDRES	14
DMEV0	CPURASERR0	SAMRRES0	15
DMEL0[2:0]	CPURASERR0	SAMRRES0	15
DMEI0[25:0]	CPURASERR0	SAMRRES0	15
DMEV1	CPURASERR1	SAMRRES1	16
DMEL1[2:0]	CPURASERR1	SAMRRES1	16
DMEI1[25:0]	CPURASERR1	SAMRRES1	16
DMEV2	CPURASERR2	SAMRRES2	17
DBE[5:0]	CPURASERR2	SAMRRES2	17
DFE[2:0]	CPURASERR2	SAMRRES2	17
SAHBWABORT	CPURASERR2	SAMRRES2	17
VMPWCA<n>[31:0] (n=0..3)	VMPARWRITE	VMPWCA<n> (n=0..3)	18
VMSCEECA<n>[31:0] (n=0..3)	VMSCECC<n> (n=0..3)	VMSCEECA<n> (n=0..3)	19+<n> (n=0..3)
VMDCEECA<n>[31:0] (n=0..3)	ECCERRVM<n> (n=0..3)	VMDUEECA<n> (n=0..3)	23+<n> (n=0..3)

4. Configuration options

The SAM provides configuration options to configure its features and components.

The following table describes the configuration options for the SAM.

Table 4-1: SAM configuration options

Name	Allowed values	Default value	Description
SAMNRA	0..7	7	Number of SAM response actions. Selects the number of response outputs implemented in the SAM. This value is reflected in SAMBC.NR. Supported values: • 0 - 1 response action • 1 – 2 response actions • 2 – 3 response actions • 3 – 4 response actions • 4 – 5 response actions • 5 – 6 response actions • 6 – 7 response actions • 7 – 8 response actions
SAMNEC	0..7	3	Number of SAM Event Counters. Selects the number of event counters implemented in the SAM. This value is reflected in SAMBC.NTC. Supported values: • 0 – 1 event counter • 1 – 2 event counters • 2 – 3 event counters • 3 – 4 event counters • 4 – 5 event counters • 5 – 6 event counters • 6 – 7 event counters • 7 – 8 event counters

Name	Allowed values	Default value	Description
SAMRESETACTION[63:0]	0, 1	0x0000_0000_0000_080D	The default behavior after reset for the event response action routing enable of each event [63:0] respectively. The possible values of each bit: • 0 – Disable routing the latched event (typically hardware default for waiting for NVM configuration loading). • 1 – Enable routing the latched event to its default response action which is typically response action 0 (Cold reset logic of the subsystem). Take into consideration: • Bit 0 is set to enable response for events from SAM configuration integrity checker. • Bit 2 is set to enable the response for events from SAM Duplication error. • If your system includes Arm's LCM, then we recommend that bit 3 and alarm input 3 are used for alarm sensing from the LCM. • If your system includes a DCLS implementation of Arm® CoreLink™ DMA-350 Controller, we recommend that bit 11 and alarm input 11 are used for alarm sensing from the DMA-350 DCLS error.
SAMEM0RVAL	32 bits	0x0000_080D	Reset value for SAMEM0 register: • Bit 0 - Enables export of the SAM Configuration Integrity Error event. • Bit 2 - Enables export of the SAM Duplication Error event. • Bit 3 - Enables export of the LCM Fatal Error event. • Bit 11 - Enables export of the DMA-350 DCLS Error event.
SAMEM1RVAL	32 bits	0x0000_0000	Reset value for SAMEM1 register
SAMIM0RVAL	32 bits	0x07FF_FFFF	Reset value for SAMIM0 register
SAMIM1RVAL	32 bits	0x0000_001F	Reset value for SAMIM1 register
SAMRRLS0RVAL	32 bits	0x0000_8828	Reset value for SAMRRLS0 register
SAMRRLS1RVAL	32 bits	0x0000_8000	Reset value for SAMRRLS1 register
SAMRRLS2RVAL	32 bits	0x0000_0000	Reset value for SAMRRLS2 register
SAMRRLS3RVAL	32 bits	0x0000_0000	Reset value for SAMRRLS3 register
SAMRRLS4RVAL	32 bits	0x0000_0000	Reset value for SAMRRLS4 register
SAMRRLS5RVAL	32 bits	0x0000_0000	Reset value for SAMRRLS5 register
SAMRRLS6RVAL	32 bits	0x0000_0000	Reset value for SAMRRLS6 register
SAMRRLS7RVAL	32 bits	0x0000_0000	Reset value for SAMRRLS7 register
SAMEC0RVAL	32 bits	0x0000_0000	Reset value for SAMEC0 register
SAMEC1RVAL	32 bits	0x0000_0000	Reset value for SAMEC1 register
SAMEC2RVAL	32 bits	0x0000_0000	Reset value for SAMEC2 register
SAMEC3RVAL	32 bits	0x0000_0000	Reset value for SAMEC3 register
SAMEC4RVAL	32 bits	0x0000_0000	Reset value for SAMEC4 register
SAMEC5RVAL	32 bits	0x0000_0000	Reset value for SAMEC5 register
SAMEC6RVAL	32 bits	0x0000_0000	Reset value for SAMEC6 register

Name	Allowed values	Default value	Description
SAMEC7RVAL	32 bits	0x0000_0000	Reset value for SAMEC7 register
SAMECTIVRVAL	32 bits	0x0000_0000	Reset value for SAMECTIV register
SAMWDCIVRVAL	32 bits	0x0000_0000	Reset value for SAMWDCIV register

5. SAM incoming events allocation

The SAM supports various internal and external alarm events.

Event 0 and event 1 are internally used by the SAM. If installed in your system, event 2 is typically used by the LCM. All other events (3:63) can be defined by the user.

For more information on LCM, see the *Arm® Lifecycle Manager Specification*.

The following table shows SAM incoming events allocation.

Table 5-1: SAM incoming events allocation

Event Index	Event Source	Signal Name	Default HW Reset Config Option (SAMRESETACTION[63:0])	Routed by NVM setup to response	Comments
0	SAM configuration integrity checker	SAM configuration integrity error	1 – Reset subsystem	0	Response Action Zero (SAMCRSTREQ) which connects at the system level to Cold reset logic of the subsystem that resets the SAM. Routed by SAMRRLS0[3:0] bits. For more information about the SAMRRLS<n> register, see 7.1.10 SAMRRLSn, Response Routing Logic register on page 45.
1	SAM watchdog timer	SAM WD interrupt	0 – Wait for NVM configuration	2	NMI to subsystem CPU[0]. Routed by SAMRRLS0[7:4] bits.
2	SAM duplication error	sam_duperr	1 – Reset subsystem	0	SAMCRSTREQ which connects at the system level to request a Cold reset which resets the SAM. Routed by SAMRRLS0[11:8] bits.
3	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS0[15:12] bits.
4	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS0[19:16] bits.
5	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS0[23:20] bits.
6	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS0[27:24] bits.
7	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS0[31:28] bits.
8	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS1[3:0] bits.
9	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS1[7:4] bits.
10	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS1[11:8] bits.

Event Index	Event Source	Signal Name	Default HW Reset Config Option (SAMRESETACTION[63:0])	Routed by NVM setup to response	Comments
11	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS1[15:12] bits.
12	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS1[19:16] bits.
13	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS1[23:20] bits.
14	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS1[27:24] bits.
15	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS1[31:28] bits.
16	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS2[3:0] bits.
17	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS2[7:4] bits.
18	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS2[11:8] bits.
19	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS2[15:12] bits.
20	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS2[19:16] bits.
21	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS2[23:20] bits.
22	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS2[27:24] bits.
23	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS2[31:28] bits.
24	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS3[3:0] bits.
25	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS3[7:4] bits.
26	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS3[11:8] bits.
27	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS3[15:12] bits.
28	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS3[19:16] bits.
29	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS3[23:20] bits.
30	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS3[27:24] bits.
31	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS3[31:28] bits.
32	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS4[3:0] bits.
33	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS4[7:4] bits.

Event Index	Event Source	Signal Name	Default HW Reset Config Option (SAMRESETACTION[63:0])	Routed by NVM setup to response	Comments
34	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS4[11:8] bits.
35	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS4[15:12] bits.
36	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS4[19:16] bits.
37	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS4[23:20] bits.
38	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS4[27:24] bits.
39	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS4[31:28] bits.
40	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS5[3:0] bits.
41	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS5[7:4] bits.
42	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS5[11:8] bits.
43	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS5[15:12] bits.
44	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS5[19:16] bits.
45	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS5[23:20] bits.
46	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS5[27:24] bits.
47	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS5[31:28] bits.
48	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS6[3:0] bits.
49	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS6[7:4] bits.
50	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS6[11:8] bits.
51	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS6[15:12] bits.
52	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS6[19:16] bits.
53	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS6[23:20] bits.
54	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS6[27:24] bits.
55	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS6[31:28] bits.
56	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS7[3:0] bits.

Event Index	Event Source	Signal Name	Default HW Reset Config Option (SAMRESETACTION[63:0])	Routed by NVM setup to response	Comments
57	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS7[7:4] bits.
58	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS7[11:8] bits.
59	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS7[15:12] bits.
60	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS7[19:16] bits.
61	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS7[23:20] bits.
62	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS7[27:24] bits.
63	Implementation defined	Implementation defined	Implementation defined	Implementation defined	Routed by SAMRRLS7[31:28] bits.

6. SAM response action allocation

The SAM supports configurable responses to alarm events.

The following table shows recommended SAM response action allocation.

Table 6-1: SAM response action allocation

Response Action ID	Action Applied	Comments
0	Sends SAMCRSTREQ signal	Typically connects in the system level to request the generation of a Cold reset. response action zero must be applied to the reset signal that resets the SAM when it detects a bad integrity check in its shadow registers to allow retry in case of intermittent error. This aligns with the reset value of SAMRRLS0 [3:0] bits that route Event 0 to response 0 and enable its routing. This signal must be cleared when its respective reset is asserted. **Note:** This action is also propagated as an output signal which may be used by the SoC level to reset the entire SoC, and to minimize the risk of unhandled pending transactions at the system level when the subsystem resets asynchronously to the system. All hardware related alarms and the asynchronous alarms should be routed to this action.
1	Sends SAMWRSTREQ signal	Typically connects in the system level to the nWARMRESETAON reset logic. This action level signal must be cleared when its respective reset is asserted. **Note:** If there's a DMA, it is notified when this response action is taken, and reset occurs only when it becomes quiescent. This reset is not likely to stall the SoC.
2	Sends NMI interrupt	Connects to a *Non-Maskable Interrupt* (NMI). This action is a level signal which is cleared when software clears its respective SAM Events Status Registers (SAMES0 and SAMES1) bit.
3	Sends critical severity fault interrupt	This action is a level signal which is cleared when software clears its respective SAM Events Status Registers (SAMES0 and SAMES1) bit.
4	Sends severe fault interrupt	This action is a level signal which is cleared when software clears its respective SAM Events Status Registers (SAMES0 and SAMES1) bit.
5	Sends SAM_ACTION_5	User defined SAM action. This action is a level signal which is cleared when software clears its respective SAM Events Status Registers (SAMES0 and SAMES1) bit. For more information about routing alarm events to this action, see Response routing logic.
6	Sends SAM_ACTION_6	User defined SAM action. This action is a level signal which is cleared when software clears its respective SAM Events Status Registers (SAMES0 and SAMES1) bit. For more information about routing alarm events to this action, see Response routing logic.
7	Sends SAM_ACTION_7	User defined SAM action. This action is a level signal which is cleared when software clears its respective SAM Events Status Registers (SAMES0 and SAMES1) bit. For more information about routing alarm events to this action, see Response routing logic.

7. Programmers model

This chapter provides general information about the SAM register properties.

The following information applies to SAM registers:

- The base address is not fixed and can be different for any particular system implementation. The offset of each register from the base address is fixed.
- Do not attempt to access reserved or unused address locations. Attempting to access these locations can result in unexpected behavior.
- Unless otherwise stated in the accompanying text:
  - Do not modify undefined register bits.
  - Ignore undefined register bits on reads.
  - All register bits are reset to the reset value specified in the register summary table of each block.

Access type is described as follows:

RW
: Read/write

RO
: Read-only

WO
: Write-only

WI
: Write ignore

RAZ
: Read as zero

RAZ/WI
: Read as zero, write ignore

RW1C
: Read write 1 to clear

RW1S
: Read write 1 to set

The SAM collects the subsystem alarm signals and issues responses as defined by its programming.

The SAM does not perform its own checks and does not filter access based on its security. Therefore, when integrating the SAM, the system must filter the access on behalf of the SAM. Arm provides a compatible Peripheral Protection Controller (PPC) that can perform this task.

We recommend that all SAM registers reside in the always-on power domain and are reset by the system Cold reset.

The following registers are reset by the PORESETn input signal:

- SAMES0
- SAMES1
- SAMCDRES
- SAMRRES0
- SAMRRES1
- SAMRRES2
- VMPWCA0
- VMPWCA1
- VMPWCA2
- VMPWCA3
- VMSCEECA0
- VMSCEECA1
- VMSCEECA2
- VMSCEECA3
- VMDUEECA0
- VMDUEECA1
- VMDUEECA2
- VMDUEECA3
- TRAMSCEECA
- TRAMDUEECA

The PORESETn signal is typically connected at the system level to the nPORESETAON signal.

The registers with offset `0x014` to `0x070` belong to the ICV protected register block. Each of these registers has a hidden shadow register which is set by a write and waits for the write to SAMICV to initiate an ICV check for the shadow registers of the block:

- If an integrity check value mismatch occurs, the SAM Configuration Integrity error alarm signal is set, and the content of the shadow registers is not copied to the functional SAM registers.
- If the ICV is correct (no mismatch), the content of the shadow registers is copied to the functional SAM registers.

A read from the ICV protected register block provides only the values of a register that were successfully integrity checked. For example, an early read of a partial write to the ICV protected register block before writing to the SAMICV register does not show the expected values in the functional registers until a successful integrity check is made.

7.1 SAM register summary

The following tables list the registers in the SAM register block.

Table 7-1: SAM register map

Offset	Name	Access	Reset value	Description
0x000	SAMBC	RO	0x0000_0000	SAM Build Configuration register. This register reflects the build configuration for software usage. Its reset value depends on the selected configuration option in Configuration options.
0x004	SAMES0	RO	0x0000_0000	SAM Event Status register 0. This register indicates the corresponding event status.
0x008	SAMES1	RO	0x0000_0000	SAM Event Status register 1. This register indicates the corresponding event status.
0x00C	SAMECL0	WO	0x0000_0000	SAM Event Clear register 0. This register provides the method to clear the corresponding event status.
0x010	SAMECL1	WO	0x0000_0000	SAM Event Clear register 1. This register provides the method to clear the corresponding event status.
0x014	SAMEM0	RW	SAMEM0RVAL	SAM Export Mask 0. This register provides the method to prevent (hide from a possible attacker) the export of selected SAM events from reaching the Status interface, which goes out to the SoC. This register is part of the registers block of the SAM.
0x018	SAMEM1	RW	SAMEM1RVAL	SAM Export Mask 1. This register provides the method to prevent (hide from a possible attacker) the export of selected SAM events from reaching the Status interface, which goes out to the SoC. This register is part of the registers block of the SAM.
0c01C	SAMIM0	RW	SAMIM0RVAL	SAM Input Mask 0 register. This register holds the Input Mask. It provides the flexibility to filter out unused input events or events from unstable or broken sensors. This register is part of the registers block of the SAM.
0x020	SAMIM1	RW	SAMIM1RVAL	SAM Input Mask 1 register. This register holds the Input Mask. It provides the flexibility to filter out unused input events or events from unstable or broken sensors. This register is part of the registers block of the SAM.
0x024	SAMRRLS0	RW	SAMRRLS0RVAL	SAM Response Routing Logic Setup register 0. This register configures the Response Routing logic. It routes each of event statuses 7:0 to one of the response actions. This register is part of the integrity checked registers block of the SAM. The reset defaults of bits 31, 27, 23, 19, 15, 11, 7, 3 are provided by the first 8 bits of SAMRESETACTION[63:0]. The reset value enable the response for events from SAM configuration integrity checker (event[0]) and SAM duplication error (event[2]) and routes them to response action 0. If LCM is also integrated, we recommend that the reset value enables also the response for the event LCM fatal error (event[2]) and routes it to response action 0. If DMA with DCLS is also integrated, we recommend that the reset value enables also the response for the event DMA DCLS error (event[1]) and routes it to response action 0.

Offset	Name	Access	Reset value	Description
0x028	SAMRRLS1	RW	SAMRRLS1RVAL	SAM Response Routing Logic Setup register 1. This register configures the Response Routing logic. It routes each of event statuses 15:8 to one of the response actions. This register is part of the registers block of the SAM. The reset defaults of bits 31, 27, 23, 19, 15, 11, 7, 3 are provided by the restpective bits ([15:8]) of SAMRESETACTION[63:0].
0x02C	SAMRRLS2	RW	SAMRRLS2RVAL	SAM Response Routing Logic Setup register 2. This register configures the Response Routing logic. It routes each of event statuses 23:16 to one of the response actions. This register is part of the registers block of the SAM. The reset defaults of bits 31, 27, 23, 19, 15, 11, 7, 3 are provided by the restpective bits ([23:16]) of SAMRESETACTION[63:0].
0x030	SAMRRLS3	RW	SAMRRLS3RVAL	SAM Response Routing Logic Setup register 3. This register configures the Response Routing logic. It routes each of event statuses 31:24 to one of the response actions. This register is part of the registers block of the SAM. The reset defaults of bits 31, 27, 23, 19, 15, 11, 7, 3 are provided by the restpective bits ([31:24]) of SAMRESETACTION[63:0].
0x034	SAMRRLS4	RW	SAMRRLS4RVAL	SAM Response Routing Logic Setup register 4. This register configures the Response Routing logic. It routes each of event statuses 39:32 to one of the response actions. This register is part of the registers block of the SAM. The reset defaults of bits 31, 27, 23, 19, 15, 11, 7, 3 are provided by the restpective bits ([39:32]) of SAMRESETACTION[63:0].
0x038	SAMRRLS5	RW	SAMRRLS5RVAL	SAM Response Routing Logic Setup register 5. This register configures the Response Routing logic. It routes each of event statuses 47:40 to one of the response actions. This register is part of the registers block of the SAM. The reset defaults of bits 31, 27, 23, 19, 15, 11, 7, 3 are provided by the restpective bits ([47:40]) of SAMRESETACTION[63:0].
0x03C	SAMRRLS6	RW	SAMRRLS6RVAL	SAM Response Routing Logic Setup register 6. This register configures the Response Routing logic. It routes each of event statuses 55:48 to one of the response actions. This register is part of the registers block of the SAM. The reset defaults of bits 31, 27, 23, 19, 15, 11, 7, 3 are provided by the first 8 bits of SAMRESETACTION[55:48].
0x040	SAMRRLS7	RW	SAMRRLS7RVAL	SAM Response Routing Logic Setup register 7. This register configures the Response Routing logic. It routes each of event statuses 63:56 to one of the response actions. This register is part of the registers block of the SAM. The reset defaults of bits 31, 27, 23, 19, 15, 11, 7, 3 are provided by the first 8 bits of SAMRESETACTION[63:56].
0x044 - 0x060	SAMECn	RW	SAMECnRVAL	SAM Event Counter register n. This register holds the value from which the event counter starts its countdown when the event that it is connected to is detected. This register is part of the integrity checked registers block of the SAM.
0x064	SAMECTIV	RW	SAMECTIVRVAL	SAM Event Counters Timer Initial Value register. This register holds the Event Counters Timer down counter initial value. This register is part of the ICV registers block of the SAM.
0x068	SAMWDCIV	RW	SAMWDCIVRVAL	SAM Watchdog Counter Initial Value register. This register holds the Watchdog down counter initial value for the Watchdog counter. This register is part of the ICV registers block of the SAM.

Offset	Name	Access	Reset value	Description
0x06C	SAMRL	RW	0x0000_0000	SAM Registers Lock register. This register provides the method to lock specific SAM registers and prevent software to reprogram them. This register is part of the ICV registers block of the SAM.
0x070	SAMICV	RW	0x0000_0000	SAM Integrity Check Value. This register protects the write of SAM registers from SAMEM0 to SAMRL with integrity check value.
0x074	SAMCDRES	RW	0x0000_0000	SAM Processor DCLS Reported Errors Status register. This register indicates the corresponding DCLS error event status.
0x078	SAMRRES0	RW	0x0000_0000	SAM Processor RAS Reported Errors Status 0 register. This register indicates the corresponding RASA error event status.
0x07C	SAMRRES1	RW	0x0000_0000	SAM Processor RAS Reported Errors Status 1 register. This register indicates the corresponding RASA error event status.
0x080	SAMRRES2	RW	0x0000_0000	SAM Processor RAS Reported Errors Status 2 register. This register indicates the corresponding RASA error event status.
0x084	VMPWCA0	RW	0x0000_0000	VM0 Partial Write Captured Address. The number of YMPWCA<n> registers depends on NUMVMBANK. If VM<n> does not exist, then VMPWCA<n> register is Reserved.
0x088	VMPWCA1	RW	0x0000_0000	VM1 Partial Write Captured Address. The number of YMPWCA<n> registers depends on NUMVMBANK. If VM<n> does not exist, then VMPWCA<n> register is Reserved.
0x08C	VMPWCA2	RW	0x0000_0000	VM2 Partial Write Captured Address. The number of YMPWCA<n> registers depends on NUMVMBANK. If VM<n> does not exist, then VMPWCA<n> register is Reserved.
0x090	VMPWCA3	RW	0x0000_0000	VM3 Partial Write Captured Address. The number of YMPWCA<n> registers depends on NUMVMBANK. If VM<n> does not exist, then VMPWCA<n> register is Reserved.
0x094	VMSCEECA0	RW	0x0000_0000	VM0 Single Corrected EEC Error Captured Address. The number of VMSCEECA<n> registers depends on NUMVMBANK. If VM<n> does not exist, then VMSCEECA<n> register is Reserved.
0x098	VMSCEECA1	RW	0x0000_0000	VM1 Single Corrected EEC Error Captured Address. The number of VMSCEECA<n> registers depends on NUMVMBANK. If VM<n> does not exist, then VMSCEECA<n> register is Reserved.
0x09C	VMSCEECA2	RW	0x0000_0000	VM2 Single Corrected EEC Error Captured Address. The number of VMSCEECA<n> registers depends on NUMVMBANK. If VM<n> does not exist, then VMSCEECA<n> register is Reserved.
0x0A0	VMSCEECA3	RW	0x0000_0000	VM3 Single Corrected EEC Error Captured Address. The number of VMSCEECA<n> registers depends on NUMVMBANK. If VM<n> does not exist, then VMSCEECA<n> register is Reserved.
0x0A4	VMDUEECA0	RW	0x0000_0000	VM0 Double Uncorrected EEC Error Captured Address. The number of VMDUEECA<n> registers depends on NUMVMBANK. If VM<n> does not exist, then VMDUCEECA<n> register is Reserved.
0x0A8	VMDUEECA1	RW	0x0000_0000	VM1 Double Uncorrected EEC Error Captured Address. The number of VMDUEECA<n> registers depends on NUMVMBANK. If VM<n> does not exist, then VMDUCEECA<n> register is Reserved.
0x0AC	VMDUEECA2	RW	0x0000_0000	VM2 Double Uncorrected EEC Error Captured Address. The number of VMDUEECA<n> registers depends on NUMVMBANK. If VM<n> does not exist, then VMDUCEECA<n> register is Reserved.
0x0B0	VMDUEECA3	RW	0x0000_0000	VM3 Double Uncorrected EEC Error Captured Address. The number of VMDUEECA<n> registers depends on NUMVMBANK. If VM<n> does not exist, then VMDUCEECA<n> register is Reserved.

Offset	Name	Access	Reset value	Description
0x0B4	TRAMSCEECA	RW	0x0000_0000	TRAM Single Corrected ECC Error Captured Address
0x0B8	TRAMDUEECA	RW	0x0000_0000	TRAM Double Uncorrected ECC Error Captured Address
0x0BC - 0xFCC	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0xFD0	PIDR4	RO	0x0000_0004	The Peripheral ID4 register returns byte[4] of the peripheral ID.
0xFD4 - 0xFDC	Reserved	**RAZ/WI**	0x0000_0000	Reserved
0xFE0	PIDR0	RO	0x0000_00F2	The Peripheral ID0 register returns byte[0] of the peripheral ID.
0xFE4	PIDR1	RO	0x0000_00B0	The Peripheral ID1 register returns byte[1] of the peripheral ID.
0xFE8	PIDR2	RO	0x0000_001B	The Peripheral ID2 register returns byte[2] of the peripheral ID.
0xFEC	PIDR3	RO	0x0000_0000	The Peripheral ID3 register returns byte[3] of the peripheral ID.
0xFF0	CIDR0	RO	0x0000_000D	The Component ID0 register returns byte[0] of the component ID.
0xFF4	CIDR1	RO	0x0000_00F0	The Component ID1 register returns byte[1] of the component ID.
0xFF8	CIDR2	RO	0x0000_0005	The Component ID2 register returns byte[2] of the component ID.
0xFFC	CIDR3	RO	0x0000_00B1	The Component ID3 register returns byte[3] of the component ID.

7.1.1 SAMBC, Build Configuration register

The SAMBC register reflects the build configuration to the software.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x000

Type

RO

Reset value

Configuration dependent

Bit descriptions

The following table shows the register bit assignments.

Table 7-2: SAMBC register bit description

Bits	Name	Description	Type	Reset
[31:17]	-	Reserved	RAZ/WI	0x0
[10:8]	NEC	Number of SAM event counters. Selects the number of event counters implemented in the SAM. Supported values are defined by SAMNEC in Configuration options.	RO	SAMNEC
[7:3]	-	Reserved	RAZ/WI	0x0
[2:0]	NRA	Number of SAM response actions. Selects the number of response outputs implemented in the SAM. Supported values are defined by SAMNRA in Configuration options.	RO	SAMNRA

7.1.2 SAMES0, Event Status register 0

The SAMES0 register indicates the status of the corresponding events from Event 0 to Event 31.

The register is reset by PORESETn signal and does not reset when the SAM is reset with the subsystem because of the alarm reset response. It holds its value to allow software to detect the reason for the alarm which resets the subsystem.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x004

Type

RO

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 7-3: SAMES0 register bit description

Bits	Name	Description	Type	Reset
[31:0]	Events	Each bit reports its corresponding event status	RO	0x0000_0000

7.1.3 SAMES1, Event Status register 1

The SAMES1 register indicates the status of the corresponding events from Event 32 to Event 63.

The register is reset by PORESETn signal and does not reset when the SAM is reset with the subsystem because of the alarm reset response. It holds its value to allow software to detect the reason for the alarm which resets the subsystem.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x008

Type

RO

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 7-4: SAMES1 register bit description

Bits	Name	Description	Type	Reset
[31:0]	Events	Each bit reports its corresponding event status	RO	0x0000_0000

7.1.4 SAMECL0, Event Clear register 0

The SAMECL0 register clears the corresponding event status for Event 0 to Event 31.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x00C

Type

WO

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 7-5: SAMECL0 register bit description

Bits	Name	Description	Type	Reset
[31:0]	Events	Each bit clears its corresponding event status in SAMES0	WO	0x0000_0000

7.1.5 SAMECL1, Event Clear register 1

The SAMECL1 register clears the corresponding event status for Event 32 to Event 63.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x010

Type

WO

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 7-6: SAMECL1 register bit description

Bits	Name	Description	Type	Reset
[31:0]	Events	Each bit clears its corresponding event status in SAMES1	WO	0x0000_0000

7.1.6 SAMEM0, Export Mask 0

The SAMEM0 register enables programing the export mask to prevent the export of selected SAM events, from 0 to 31. The selected SAM events cannot reach the status interface which goes out to the rest of the host system.

The SAMEM0 has a shadow register and functional register. When written, the write value is held in the shadow register until SAMICV is written with correct ICV value. Only then a read shows the written value in the actual register.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

`0x014`

Type

The SAMEM0 register is RW while it is not locked, as defined by SAMRL. L_SAMEM0, otherwise it is **RAZ/WI**.

Reset value

SAMEM0RVAL

Bit descriptions

The following table shows the register bit assignments.

Table 7-7: SAMEM0 register bit description

Bits	Name	Description	Type	Reset
[31:0]	Events	Each bit enables the export of its corresponding event. This register masks events 0 to 31.	RW	SAMEM0RVAL

7.1.7 SAMEM1, Export Mask 1

The SAMEM1 register enables programing the export mask to prevent the export of selected SAM events, from 32 to 63. The selected SAM events cannot reach the status interface which goes out to the rest of the host system.

The SAMEM1 has a hidden shadow register and functional register. When written, the write value is held in the shadow register until SAMICV is written with correct ICV value. Only then a read would show the written value in the actual register.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

`0x018`

Type

The SAMEM1 register is RW while it is not locked, as defined by SAMRL. L_SAMEM1, otherwise it is **RAZ/WI**.

Reset value

SAMEM1RVAL

Bit descriptions

The following table shows the register bit assignments.

Table 7-8: SAMEM1 register bit description

Bits	Name	Description	Type	Reset
[31:0]	Events	Each bit enables the export of its corresponding event. This register masks events 32 to 63.	RW	SAMEM1RVAL

7.1.8 SAMIM0, Input Mask 0 register

The SAMIM0 register specifies which of the inputs for the corresponding events, from event 0 to event 31, are filtered out.

The SAMIM0 has a hidden shadow register and functional register. When written, the write value is held in the shadow register until SAMICV is written with the correct ICV value. Only then a read would show the written value in the functional register.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x01C

Type

The SAMIM0 is RW while it is not locked, per SAMRL.L_SAMIM0, otherwise it is **RAZ/WI**.

Reset value

SAMIM0RVAL

Bit descriptions

The following table shows the register bit assignments.

Table 7-9: SAMIM0 register bit description

Bits	Name	Description	Type	Reset
[31:0]	Events	Each bit masks its corresponding input event. When the bit is 0 then the event is masked out.	RW	SAMIM0RVAL

7.1.9 SAMIM1, Input Mask 1 register

The SAMIM1 register specifies which of the inputs for the corresponding events, from event 32 to event 63, are filtered out.

The SAMIM1 has a hidden shadow register and functional register. When written, the write value is held in the shadow register until SAMICV is written with correct ICV value. Only then a ead would show the written value in the functional register.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x020

Type

The SAMIM1 register is RW while it is not locked, per SAMRL.L_SAMIM1, otherwise it is **RAZ/WI**.

Reset value

SAMIM1RVAL

Bit descriptions

The following table shows the register bit assignments.

Table 7-10: SAMIM1 register bit description

Bits	Name	Description	Type	Reset
[31:0]	Events	Each bit masks its corresponding input event. When the bit is 0, the event is masked out.	RW	SAMIM1RVAL

7.1.10 SAMRRLSn, Response Routing Logic register

The SAMRRLSn register (n=0..7) configures the response routing logic.

When written, the write value is held in the shadow register until SAMICV is written with correct ICV value. Only then a read would show the written value in the functional register.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x024 - 0x040

Type

RW while it is not locked, as defined by SAMRL.L_SAMRRLS<n>, otherwise it is **RAZ/WI**. The SAMRRLSn has a hidden shadow register and a functional register.

Reset value

SAMRRLS<n>RVAL

Bit descriptions

The following table shows the register bit assignments.

Table 7-11: SAMRRLSn register bit description

Bits	Name	Description	Type	Reset
[31]	ERARE[8n+7]	Event Response Action Routing Enable. When set to 1, enables the routing of Event [8n+7] to its allocated action. Otherwise, the event latched in SAMES remains shadow. The default value of this bit is defined by its respective bit in SAMRRLS<n>RVAL configuration option.	RW	0x0
[30:28]	ERAR[8n+7]	Event Response Action Routing. Identifies to which of the Response Actions (0..7), Event [8n+7] latched in the event status registers is routed. The default value of this bit is defined by its respective bit in SAMRRLS<n>RVAL configuration option.	RW	0x0
[27]	ERARE[8n+6]	Event Response Action Routing Enable. When set to 1, enables the routing of Event [8n+6] to its allocated action. Otherwise, the event latched in SAMES remains shadow. The default value of this bit is defined by its respective bit in SAMRRLS<n>RVAL configuration option.	RW	0x0

Bits	Name	Description	Type	Reset
[26:24]	ERAR[8n+6]	Event Response Action Routing. Identifies to which of the Response Actions (0..7), Event [8n+6] latched in the event status registers is routed. The default value of this bit is defined by its respective bit in SAMRRLS<n>RVAL configuration option.	RW	0x0
[23]	ERARE[8n+5]	Event Response Action Routing Enable. When set to 1, enables the routing of Event [8n+5] to its allocated action. Otherwise, the event latched in SAMES remains shadow. The default value of this bit is defined by its respective bit in SAMRRLS<n>RVAL configuration option.	RW	0x0
[22:20]	ERAR[8n+5]	Event Response Action Routing. Identifies to which of the Response Actions (0..7), Event [8n+5] latched in the event status registers is routed. The default value of this bit is defined by its respective bit in SAMRRLS<n>RVAL configuration option.	RW	0x0
[19]	ERARE[8n+4]	Event Response Action Routing Enable. When set to 1, enables the routing of Event [8n+4] to its allocated action. Otherwise, the event latched in SAMES remains shadow. The default value of this bit is defined by its respective bit in SAMRRLS<n>RVAL configuration option.	RW	0x0
[18:16]	ERAR[8n+4]	Event Response Action Routing. Identifies to which of the Response Actions (0..7), Event [8n+4] latched in the event status registers is routed. The default value of this bit is defined by its respective bit in SAMRRLS<n>RVAL configuration option.	RW	0x0
[15]	ERARE[8n+3]	Event Response Action Routing Enable. When set to 1, enables the routing of Event [8n+3] to its allocated action. Otherwise, the event latched in SAMES remains shadow. The default value of this bit is defined by its respective bit in SAMRRLS<n>RVAL configuration option.	RW	0x0
[14:12]	ERAR[8n+3]	Event Response Action Routing. Identifies to which of the Response Actions (0..7), Event [8n+3] latched in the event status registers is routed. The default value of this bit is defined by its respective bit in SAMRRLS<n>RVAL configuration option.	RW	0x0
[11]	ERARE[8n+2].	Event Response Action Routing Enable. When set to 1 enables the routing of Event [8n+2] to its allocated action. Otherwise, the event latched in SAMES remains shadow. The default value of this bit is defined by its respective bit in SAMRRLS<n>RVAL configuration option.	RW	0x0
[10:8]	ERAR[8n+2]	Event Response Action Routing. Identifies to which of the Response Actions (0..7), Event [8n+2] latched in the event status registers is routed. The default value of this bit is defined by its respective bit in SAMRRLS<n>RVAL configuration option.	RW	0x0
[7]	ERARE[8n+1]	Event Response Action Routing Enable. When set to 1 enables the routing of Event [8n+1] to its allocated action. Otherwise, the event latched in SAMES remains shadow. The default value of this bit is defined by its respective bit in SAMRRLS<n>RVAL configuration option.	RW	0x0
[6:4]	ERAR[8n+1]	Event Response Action Routing. Identifies to which of the Response Actions (0..7), Event [8n+1] latched in the event status registers is routed. The default value of this bit is defined by its respective bit in SAMRRLS<n>RVAL configuration option.	RW	0x0
[3]	ERARE[8n+0]	Event Response Action Routing Enable. When set to 1 enables the routing of Event [8n+0] to its allocated action. Otherwise, the event latched in SAMES remains shadow. The default value of this bit is defined by its respective bit in SAMRRLS<n>RVAL configuration option.	RW	0x0

Bits	Name	Description	Type	Reset
[2:0]	ERAR[8n+0]	Event Response Action Routing. Identifies to which of the Response Actions (0..7), Event [8n+0] latched in the event status registers is routed. The default value of this bit is defined by its respective bit in SAMRRLS<n>RVAL configuration option.	RW	0x0

7.1.11 SAMECn SAM Event Counter register

The SAMECn (n=0..7) register configures a counter that is used to filter the input event to make it immune to noise. Prevents the action of certain events until the counter reaches zero.

This register has a hidden shadow register and a functional register. When written, the write value is held in the shadow register until SAMICV is written with correct ICV value. Only then a read would show the written value in the actual register.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x044 - 0x060

Type

RW while it is not locked (as defined by SAMRL.L_SAMEC<n>), otherwise this register is **RAZ/WI**. If the event counter is not implemented (the value of SAMNEC is < n) then register is **RAZ/WI**.

Reset value

SAMECnRVAL

Bit descriptions

The following table shows the register bit assignments.

Table 7-12: SAMECn register bit description

Bits	Name	Description	Type	Reset
[31:16]	Count	The counter value. If the initial value is set to 0, the event counter does not count down and does not filter the incoming event or allow the events to be passed through. The default value of these bits is as defined by its respective bits in SAMEC<n>RVAL.	RW	0x0
[15:7]	-	Reserved	**RAZ/WI**	0x0

Bits	Name	Description	Type	Reset
[6]	Enable	When set to 1, this event counter is enabled and engaged. The event counter being engaged means it prevents events from the pointed event from propagating to the SAMES0 or SAMES1 while the counter is not 0. When the counter is enabled, it counts down every time the selected event occurs. While the count is nonzero, it blocks the event from passing through to be reach the status registers. The next event passes in and is registered only when the counter reaches zero. When set to 0, this event counter is not used, which means it does not prevent any of the events. The default value of this bit is defined by its respective bits in SAMEC<n>RVAL.	RW	0x0
[5:0]	Event	Identifies which of the input events this event counter is connected to (0..63). The default value of these bits is defined by its respective bits in SAMEC<n>RVAL.	RW	0x0

7.1.12 SAMECTIV, Event Counters Timer Initial Value register

The SAMECTIV register holds the Event Counters Timer down counter initial value. The register is part of the ICV registers block of the SAM.

The SAMECTIV has a hidden shadow register and a functional register. When written, the write value is held in the shadow register until SAMICV is written with correct ICV value. Only then a read would show the written value in the functional register.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x064

Type

RW while it is not locked, as defined by SAMRL.L_SAMECTIV, otherwise it is **RAZ/WI**.

Reset value

SAMECTIVRVAL

Bit descriptions

The following table shows the register bit assignments.

Table 7-13: SAMECTIV register bit description

Bits	Name	Description	Type	Reset
[31:0]	IV	*Initialization Value* (IV) for the Event Counters Timer. When set to 0, the Event Counters Timer does not count. Therefore, it does not expire.	RW	SAMECTIVRVAL

7.1.13 SAMWDCIV, Watchdog Counter Initial Value register

The SAMWDCIV register defines the watchdog counter down count value and which software related response action triggers the watchdog.

The SAMWDCIV has a hidden shadow register and a functional register. When written, the write value is held in the shadow register until SAMICV is written with correct ICV value. Only then a read would show the written value in the functional register.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x068

Type

RW while it is not locked, as defined by SAMRL.L_SAMWDCIV, otherwise it is **RAZ/WI**.

Reset value

SAMWDCIVRVAL

Bit descriptions

The following table shows the register bit assignments.

Table 7-14: SAMWDCIV register bit description

Bits	Name	Description	Type	Reset
[31]	RA7M	Response Action 7 Mask. When set to 1, when response action 7 is set to 1 the watchdog is fired and starts it countdown. See the allocation of the response actions. The default value of this field is defined by its respective bits in SAMWDCIVRVAL.	RW	0
[30]	RA6M	Response Action 6 Mask. When set to 1, when response action 6 is set to 1 the watchdog is fired and starts it countdown. See the allocation of the response actions. The default value of this field is defined by its respective bits in SAMWDCIVRVAL.	RW	0

Bits	Name	Description	Type	Reset
[29]	RA5M	Response Action 5 Mask. When set to 1, when response action 5 is set to 1 the watchdog is fired and starts it countdown. See the allocation of the response actions. The default value of this field is defined by its respective bits in SAMWDCIVRVAL.	RW	0
[28]	RA4M	Response Action 4 Mask. When set to 1, when response action 4 (usually - Severe Fault Interrupt) is set to 1 the watchdog is fired and starts it countdown. See the allocation of the response actions. The default value of this field is defined by its respective bits in SAMWDCIVRVAL.	RW	0
[27]	RA3M	Response Action 3 Mask. When set to 1, when response action 3 (usually - Critical Severity Fault Interrupt) is set to 1 the watchdog is fired and starts it countdown. See the allocation of the response actions. The default value of this field is defined by its respective bits in SAMWDCIVRVAL.	RW	0
[26]	RA2M	Response Action 2 Mask. When set to 1, when response action 2 (usually - NMI) is set to 1 the watchdog is fired and starts it countdown. See the allocation of the response actions. The default value of this field is defined by its respective bits in SAMWDCIVRVAL.	RW	0
[25:0]	IV	Initialization value for the Watchdog Counter. When set to 0, the Watchdog Counter does not count, therefore it does not expire. The default value of this field is defined by its respective bits in SAMWDCIVRVAL.	RW	0

7.1.14 SAMRL, Registers Lock register

The SAMRL register locks each of the related SAM registers for write. A write to a locked register's shadow register is not propagated to its functional register even if the integrity check on the ICV protected register block is successful.

When written, the write value is held in the shadow register until SAMICV is written with correct ICV value. Only then a read would show the written value in the functional register.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x06C

Type

RW while it is not locked, as defined by SAMRL.L_SAMRL, otherwise it is **RAZ/WI**. The SAMRL has a hidden shadow register and a functional register.

If an event counter is not implemented (SAMNEC configuration option< n), the respective (L_SAMECn) bit is **RAZ/WI**.

For example, If the event counter is not implemented (the value of SAMNEC < 3), the L_SAMEC3 bit is **RAZ/WI**.

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 7-15: SAMRL register bit description

Bits	Name	Description	Type	Reset
[31:28]	Reserved	-	RO	0
[27]	L_SAMRL	Allows Secure software to lock the SAMRL register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[26]	L_SAMWDCIV	Allows Secure software to lock the SAMWDCIV register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[25]	L_SAMECTIV	Allows Secure software to lock the SAMECTIV register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[24]	L_SAMEC7	Allows Secure software to lock the SAMEC7 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[23]	L_SAMEC6	Allows Secure software to lock the SAMEC6 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[22]	L_SAMEC5	Allows Secure software to lock the SAMEC5 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[21]	L_SAMEC4	Allows Secure software to lock the SAMEC4 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[20]	L_SAMEC3	Allows Secure software to lock the SAMEC3 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[19]	L_SAMEC2	Allows Secure software to lock the SAMEC2 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[18]	L_SAMEC1	Allows Secure software to lock the SAMEC1 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM. .	RW1S	0x0
[17]	L_SAMEC0	Allows Secure software to lock the SAMEC0 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[16]	L_ SAMRRLS7	Allows Secure software to lock the SAMRRLS7 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[15]	L_ SAMRRLS6	Allows Secure software to lock the SAMRRLS6 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[14]	L_ SAMRRLS5	Allows Secure software to lock the SAMRRLS5 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[13]	L_ SAMRRLS4	Allows Secure software to lock the SAMRRLS4 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[12]	L_ SAMRRLS3	Allows Secure software to lock the SAMRRLS3 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0

Bits	Name	Description	Type	Reset
[11]	L_ SAMRRLS2	Allows Secure software to lock the SAMRRLS2 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[10]	L_ SAMRRLS1	Allows Secure software to lock the SAMRRLS1 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[9]	L_SAMRRLS0	Allows Secure software to lock the SAMRRLS0 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[8]	L_SAMIM1	Allows Secure software to lock the SAMIM1 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[7]	L_SAMIM0	Allows Secure software to lock the SAMIM0 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[6]	L_SAMEM1	Allows Secure software to lock the SAMEM1 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[5]	L_SAMEM0	Allows Secure software to lock the SAMEM0 register for further writes. This security capability prevents malicious software from harming the functionality of the SAM.	RW1S	0x0
[4:0]	Reserved	-	RO	0

7.1.15 SAMICV, Integrity Check Value

The SAMICV is the last register that the software or an external entity like a DMA writes to when it updates its ICV protected register block.

The integrity check value written to the SAMICV register is the zero count over all the ICV protected register block. When this register is written, the SAM configuration checker reads the respective shadow registers of the ICV protected register block and computes the integrity check value on their values.

- If the ICV does not match, alarm signal is set, and the content of the shadow registers is not copied to the SAM operational registers.
- If the ICV does match, the content of the shadow registers is copied to the SAM registers.

This register has a hidden shadow register and a functional register. When written, the write value is held in the shadow register until SAM determines that the ICV is written with correct ICV value. Only then a read shows the written value in the actual register.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x070

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 7-16: SAMICV register bit description

Bits	Name	Description	Type	Reset
[31:10]	-	-	RAZ/WI	0x0000_0000
[9:0]	ICV	*Integrity Check Value* (ICV) for the ICV protected register block registers (not including this register). When the SAM setup is held in OTP, the preferred integrity check value is zero count.	RW	0x0000_0000

7.1.16 SAMCDRES, Processor DCLS Reported Errors Status register

The SAMCDRES register captures and holds the values of processor DCLS asserted signals when input event 14 is set.

When software wants to acknowledge a captured event in the SAMCDRES register, it should:

1. Clear its source
2. Write 1 to the error respective bit in SAMCDRES register.
3. If there is a new input event after clearing the event source and before software clears its respective bit in SAMCDRES, the status bit remains set.
4. Software must then clear the CPUDCLSERR event bit in SAMES0 register by setting the respective bit in SAMECL0 register.

   If Dual-Core Lock-Step (DCLS) is integrated, the SAM response typically resets which clears the DCLS error signals, therefore software has nothing to clear.

This register resets by the PORESETn signal and is not reset when the SAM is reset with the subsystem due to alarm reset response. It holds its value to allow software to detect the reason for the alarm which reset the subsystem.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x074

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 7-17: SAMCDRES register bit description

Bits	Name	Description	Type	Reset
[31:18]	Reserved	Reserved	**RAZ/WI**	0x0
[17:12]	DCLSIWICCOMPRES	Captures DCLSIWICCOMPRES[5:0] signals from the processor	RW1C	0x0
[11:0]	DCLSCORECOMPRES	Captures DCLSCORECOMPRES[11:0] signals from the processor	RW1C	0x0

7.1.17 SAMRRES0, Processor RAS Reported Error Status register

The SAMRRES0 register captures the RAS asserted signals and holds their values when input event 15 is set.

When software acknowledges a captured event in the SAMRRES0 register, it should:

1. Clear its source
2. Write 1 to the error respective bit in the SAMRRES0 register.
3. If there is a new input event after clearing the event source and before software clears its respective bit in SAMRRES0, the status bit remains set.
4. Then the software must clear the CPURASERR event bit in SAMES0 register by setting the respective bit in SAMECL0 register.

The register resets by the PORESETn signal and is not reset when the SAM is reset with the subsystem due to alarm reset response. It holds its value to allow software to detect the reason for the alarm which reset the subsystem.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x078

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 7-18: SAMRRES0 register bit description

Bits	Name	Description	Type	Reset
[31:30]	Reserved	Reserved	RAZ/WI	0x0
[29:4]	DMEI0	Captures DMEI0[25:0] signals from the processor	RWIC	0x0
[3:1]	DMELO	Captures DMEL0[2:0] signals from the processor	RWIC	0x0
[0]	DMEVO	Captures DMEV0 signals from the processor	RWIC	0x0

7.1.18 SAMRRES1, Processor RAS Reported Error Status register

The SAMRRES1 register captures the RAS asserted signals and holds their values when input event 16 is set.

When software acknowledges a captured event in the SAMRRES1 register, it should:

1. Clear its source
2. Write 1 to the error respective bit in the SAMRRES1 register.
3. If there is a new input event after clearing the event source and before software clears its respective bit in SAMRRES1, the status bit remains set.
4. The software must clear the CPURASERR event bit in SAMES0 register by setting the respective bit in SAMECL0 register.

The register resets by the PORESETn signal and is not reset when the SAM is reset with the subsystem due to alarm reset response. It holds its value to allow software to detect the reason for the alarm which reset the subsystem.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x07C

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 7-19: SAMRRES1 register bit description

Bits	Name	Description	Type	Reset
[\31:30]	Reserved	Reserved	**RAZ/WI**	0x0
[29:4]	DMEI1	Captures DMEI1[25:0] signals from the processor	RWIC	0x0
[3:1]	DMEL1	Captures DMEL1[2:0] signals from the processor	RWIC	0x0
[0]	DMEV1	Captures DMEV1 signals from the processor	RWIC	0x0

7.1.19 SAMRRES2, Processor RAS Reported Error Status register

The SAMRRES2 register captures the RAS asserted signals and holds their values when input event 17 is set. Additionally, the register also hosts the SAHBWABORT which is not purely categorized as RAS but serves a similar purpose.

When software acknowledges a captured event in the SAMRRES2 register, it should:

1. Clear its source
2. Write 1 to the error respective bit in the SAMRRES2 register.
3. If there is a new input event after clearing the event source and before software clears its respective bit in SAMRRES2, the status bit remains set.
4. The software must clear the CPURASERR event bit in SAMES0 register by setting the respective bit in SAMECL0 register.

In the case of RAS, the SAM response is typically reset which clears the RAS error signals, therefore software has nothing to clear.

The register resets by the PORESETn signal and is not reset when the SAM is reset with the subsystem due to alarm reset response. It holds its value to allow software to detect the reason for the alarm which reset the subsystem.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x080

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 7-20: SAMRRES2 register bit description

Bits	Name	Description	Type	Reset
[31:11]	Reserved	Reserved	**RAZ/WI**	0x0
[10]	SAHBWABORT	Captures SAHBWABORT signals from the processor	RWIC	0x0
[9:7]	DFE	Captures DFE[2:0] signals from the processor	RWIC	0x0
[6:1]	DBE	Captures DBE[5:0] signals from the processor	RWIC	0x0
0	DMEV2	Captures DMEV2 signals from the processor	RWIC	0x0

7.1.20 VMPWCA<n>, Volatile Memory Partial Write Captured Address register

The VMPWCA<n> register latches the latest address violating subsystem volatile memories (SRAMs) two words write access requirement by performing partial write. Software can clear the register after reading it to make it ready for the next possible detection.

SRAMs are arranged in lines of 64 bits wide and are protected by ECC for every line. Partial writes to a line are not supported and both software and DMA must access full lines. In the case of a violation, the written data to the SRAM is corrupted. Alarm signal PARWRVM<n> is set and is combined with the other PARWRVM<n> signals to VMPARWRITE input signal to the SAM. The offending address is captured in this register when input event 18 is set. It allows software to know to which address a partial write occured in the respective VM<n> (n=0..3).

This register resets by the the PORESETn signal and is not reset when the SAM is reset with the subsystem due to alarm reset response. It holds its value to allow software to detect the reason for the alarm which reset the subsystem.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x084 - 0x090

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 7-21: VMPWCA<n> register bit description

Bits	Name	Description	Type	Reset
[31:0]	VMPWCA	VM Partial Write Captured Address	RW	0x0

7.1.21 VMSCEECA<n>, Volatile Memory Single Corrected ECC Error Captured Address register

The VMSCEECA<n> register latches the latest corrected address when the respective input event (19, 20, 21, or 22) is set to 1.

When trying to recover an SRAM entry which resulted in a single corrected ECC error, software can perform a read operation. This can result in a new, corrected data if the error is persistent, or good data if the previous read was temporarily corrupted and correctable. Then Software writes back the same data to the same SRAM address in an attempt to fix a persistent error. Software can zeroize this register after reading it to make it ready for the next possible detection.

The subsystem volatile memories (SRAMs) are arranged in lines of 64 bits wide and are protected by ECC for every line. During a read operation the SRAM checks the ECC and in the case of a single error it provides the corrected value to the reader. Interrupt signal SCECCVM<n> is set and is combined with other SCECCVM<n> interrupt signals to a combined interrupt signal which sets IRQ[37] to software. The address of the corrected data is captured in this register. It allows software to know from which address the corrected data was read from the respective VM<n> (n=0..3).

A single corrected ECC error generates an interrupt and does not generate SAM event, so it does not reset the subsystem. This register resets by the PORESETn signal and is not reset when the SAM is reset with the subsystem due to an alarm reset response. Therefore the register holds its value to allow software to detect a single corrected ECC error address which was captured before a later SAM reset event occured.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

`0x094 - 0x0A0`

Type

RW

Reset value

`0x0000_0000`

Bit descriptions

The following table shows the register bit assignments.

Table 7-22: VMSCEECA<n> register bit description

Bits	Name	Description	Type	Reset
[31:0]	VMSCEECA	VM Single Corrected ECC Error Captured Address	RW	0x0

7.1.22 VMDUEECA<n>, Volatile Memory Double Uncorrected ECC Error Captured Address register

The VMDUEECA<n> register latches the latest uncorrected SRAM read address when the respective input event (23, 24, 25, or 26) is set. Software can zeroize the register after reading it to make it ready for the next possible detection.

The subsystem volatile memories (SRAMs) are arranged in lines of 64 bits wide and are protected by ECC for every line. In reads the SRAM checks the ECC and in the case of a double error it provides an uncorrected value to the reader. Alarm signal ECCERRVM<n> is set and is combined with the other ECCERRVM<n> signals to ECCERRSRAMS input signal to the SAM. The address of the uncorrected data is captured in this register. It allows software to know from which address the uncorrected data was read from the respective VM<n> (n=0..3).

This register resets by PORESETn signal and is not reset when the SAM is reset with the subsystem due to alarm reset response. It holds its value to allow software to detect the reason for the alarm which reset the subsystem.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

`0x0A4 - 0x0B0`

Type

RW

Reset value

`0x0000_0000`

Bit descriptions

The following table shows the register bit assignments.

Table 7-23: VMSCEECA<n> register bit description

Bits	Name	Description	Type	Reset
[31:0]	VMDUEECA	VM Double Uncorrected ECC Error Captured Address	RW	0x0

7.1.23 TRAMSCEECA, TRAM Single Corrected ECC Error Captured Address register

The TRAMSCEECA register captures the address of the corrected data of the combined interrupt signal that sets IRQ[37] to the software. It allows software to know from which address the corrected data was read from the TRAM.

The subsystem TRAM is arranged in lines of 64 bits wide and are protected by ECC for every byte. In reads the TRAM checks the ECC and in the case of a single error it provides corrected value to the reader. Software can perform a read operation and write back the same data to the same TRAM address in attempt to fix a persistent error. Software can zeroize this register after reading it to make it ready for a possible next detection. Software performing the read can result with new, corrected data if the error is persistent, or good data if the previous read was temporarily corrupted and correctable.

A single corrected ECC error generates an interrupt and does not generate SAM event, therefore it does not reset the subsystem. This register resets by n_event_status_reset and is not reset when

the SAM is reset with the subsystem due to an alarm reset response. Therefore, this register holds its value to allow software to detect a single corrected ECC error address that was captured before a later SAM reset event occurred.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x0B4

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 7-24: TRAMSCEECA register bit description

Bits	Type	Default	Name	Description
[31:0]	RW	0x0000_0000	TRAMSCEECA	TRAM Single Corrected ECC Error Captured Address.

7.1.24 TRAMDUEECA, TRAM Double Uncorrected ECC Error Captured Address register

The TRAMDUEECA register captures the address of the uncorrected data of the double error which TRAM can encounter while checking the ECC during read. Capturing the address allows software to know from which address the uncorrected data was read from the TRAM.

The subsystem TRAM is arranged in lines of 64 bits wide and are protected by ECC for every byte. The alarm signal eccerrtram is set and is combined with all the eccerrsram<n> signals to ECCERRSRAMS input the to the SAM. This register latches the latest uncorrected address. Software can zeroize this register after reading it to make it ready for the next possible detection. The TRAMDUEECA register resets by the n_event_status_reset signal and is not reset when the SAM is reset with the subsystem due to alarm reset response. It holds its value to allow software to detect the reason for the alarm which resets the subsystem.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0x0B8

Type

RW

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 7-25: TRAMDUEECA register bit description

Bits	Type	Default	Name	Description
[31:0]	RW	0x0000_0000	TRAMDUEECA	TRAM Double Uncorrected ECC Error Captured Address

7.1.25 PIDR4, Peripheral ID 4 register

The Peripheral ID4 register returns byte[4] of the peripheral ID.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0xFD0

Type

RO

Reset value

0x0000_0004

Bit descriptions

The following table shows the register bit assignments.

Table 7-26: Peripheral ID 4 register bit assignments

Bits	Name	Description	Type	Reset
[31:8]	-	Reserved	-	-
[7:4]	SIZE	4KB Count - the number of 4K pages used: • `0x00`: 4K • `0x01`: 8K • `0x02`: 16K • `0x03`: 32K	RO	`0x0`
[3:0]	DES_2	JEP106 Continuation Code	RO	`0x4`

7.1.26 PIDR0, Peripheral ID 0 register

The Peripheral ID0 register returns byte[0] of the peripheral ID.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

`0xFE0`

Type

RO

Reset value

`0x0000_00F2`

Bit descriptions

The following table shows the register bit assignments.

Table 7-27: Peripheral ID 0 register bit assignments

Bits	Name	Description	Type	Reset
[31:8]	-	Reserved	-	`0x00`
[7:0]	Peripheral ID 0	PART_0 - Identification register part number, bits[7:0]	RO	`0xF2`

7.1.27 PIDR1, Peripheral ID 1 register

The Peripheral ID1 register returns byte[1] of the peripheral ID.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

`0xFE4`

Type

RO

Reset value

`0x0000_00B0`

Bit descriptions

The following table shows the register bit assignments.

Table 7-28: Peripheral ID 1 register bit assignments

Bits	Name	Description	Type	Reset
[31:8]	-	Reserved	-	-
[7:4]	DES_0	JEP106 identification code, bits[3:0].	RO	`0xB`
[3:0]	PART_1	Identification register part number, bits[11:8]	RO	`0x0`

7.1.28 PIDR2, Peripheral ID 2 register

The Peripheral ID2 register returns byte[2] of the peripheral ID.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0xFE8

Type

RO

Reset value

0x0000_001B

Bit descriptions

The following table shows the register bit assignments.

Table 7-29: Peripheral ID 2 register bit assignments

Bits	Name	Description	Type	Reset
[31:8]	_	Reserved	_	_
[7:4]	REVISION	Revision Code	RO	0x1
[3]	JEDEC	JEDEC	RO	0x1
[2:0]	DES_1	JEP106 identification code, bits[6:4].	RO	0x3

7.1.29 PIDR3, Peripheral ID 3 register

The Peripheral ID3 register returns byte[3] of the peripheral ID.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0xFEC

Type

RO

Reset value

0x0000_0000

Bit descriptions

The following table shows the register bit assignments.

Table 7-30: Peripheral ID 3 register bit assignments

Bits	Name	Description	Type	Reset
[31:8]	-	Reserved	-	-
[7:4]	REVAND	Manufacturer revision number	RO	0x0
[3:0]	CMOD	Customer Modified	RO	0x0

7.1.30 CIDR0, Component ID 0 register

The Component ID0 register returns byte[0] of the component ID.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0xFF0

Type

RO

Reset value

0x0000_000D

Bit descriptions

The following table shows the register bit assignments.

Table 7-31: Component ID 0 register bit assignments

Bits	Name	Description	Type	Reset
[31:8]	-	Reserved	-	-
[7:0]	PRMBL_0	Preamble	RO	0xD

7.1.31 CIDR1, Component ID 1 register

The Component ID1 register returns byte[1] of the component ID.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0xFF4

Type

RO

Reset value

0x0000_00F0

Bit descriptions

The following table shows the register bit assignments.

Table 7-32: Component ID 1 register bit assignments

Bits	Name	Description	Type	Reset
[31:8]	-	Reserved	-	-
[7:4]	CLASS	Component class	RO	0xF
[3:0]	PRMBL_1	Preamble	RO	0x0

7.1.32 CIDR2, Component ID 2 register

The Component ID2 register returns byte[2] of the component ID.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0xFF8

Type

RO

Reset value

0x0000_0005

Bit descriptions

The following table shows the register bit assignments.

Table 7-33: Component ID 2 register bit assignments

Bits	Name	Description	Type	Reset
[31:8]	-	Reserved	-	-
[7:0]	PRMBL_2	Preamble	RO	0x5

7.1.33 CIDR3, Component ID 3 register

The Component ID3 register returns byte[3] of the component ID.

Configurations

This register is available in all configurations.

Attributes

Width

32

Functional group

SAM register summary

Address offset

0xFFC

Type

RO

Reset value

0x0000_00B1

Bit descriptions

The following table shows the register bit assignments.

Table 7-34: Component ID 3 register bit assignments

Bits	Name	Description	Type	Reset
[31:8]	-	Reserved	-	-
[7:0]	PRMBL_3	Preamble	RO	0xB1

Appendix A Revisions

The following table describes the technical changes between released issues of this document.

Table A-1: Issue 0000-01

Change	Location
Initial issue of the document	-