7.8.2.1 Registers

Instances

Instance	Domain	Base address	TrustZone			Split access	Description
			Map	Att	DMA		
GLITCHDET	GLOBAL	0x5004B000	HF	S	NA	No	Glitch detectors

Register overview

Register	Offset	TZ	Description
CONFIG	0x5A0		Configuration for glitch detector

7.8.2.1.1 CONFIG

Address offset: 0x5A0

Configuration for glitch detector

Bit number																															
ID																														B	A
Reset 0x00000001																														**0**	**1**
ID	R/W	Field		Value ID		Value																									
A	RW	ENABLE																													
				Disable		0																									
				Enable		1																									
B	RW	MODE																													
				HighPassFilter		0																									
				CapDiv		1																									

7.8.3 KMU — Key management unit

The key management unit (KMU) provides secure key storage functions by storing data in a dedicated region of RRAM.

The secure information configuration region, SICR, is the RRAM region that holds keys seeds, and metadata. Access to KMU and the key slots in SICR is only allowed from secure mode. KMU has exclusive access to SICR, meaning the rest of the system does not have access. The KMU stores data in key slots that hold one 128-bit value together with an access policy and a destination address for the key value. Multiple key slots can be combined to hold key sizes larger than 128 bits. How and when a key value can be used is determined by the access policy. When requested by the CPU, the destination address, which is part of the key slot, determines the memory map location for the key value that is pushed by KMU.

Key slots can be configured to be pushed directly into write-only key registers or RAM of cryptographic accelerators like CRACEN, without revealing the key value to the CPU. This enables the CPU to use the key values stored inside the key slots for cryptographic operations without knowing the key value.

KMU can also store other secrets, like the CRACEN SEED register, using multiple key slots.

A good design practice is to overwrite previously pushed secrets when they are no longer in use. KMU can be used for this, by pushing key slots with previously generated random data to the key RAM.