Hello, and welcome to this presentation of the embedded Flash memory which is included in all products of the STM32H5 microcontroller Series.

FLASH features

Feature	STM32H5
Maximum size*	Up to 2 MB
Number of banks	2
Sector size	8 Kbytes
Read data bus width	128 bits
Endurance (program/erase)	10 Kcycles 100 Kcycles on high cycle data area
One-Time-Programming	2 Kbytes
Prefetch	✓
Bank swapping	✓
Device life cycle	✓

* depends on product

life.augmented

2

This table summarizes the features of the flash memory existing in STM32H5.
Depending on the product, the flash size can be up to 2MB. It also embeds a one-time-programming area of 2 kilobytes.
The flash read data bus width is 128-bit. STM32H5 always supports a dual bank architecture. The SWAP-BANK option in the user option bytes is used to swap Bank 1 and Bank 2 addresses.
As the nonvolatile memory is divided into two independent banks, the embedded flash memory interface supports a read in one bank while a write or an erase is executed in the other bank.
The sector size which provides the minimum erase

granularity is 8 kilobytes.
Flash memory may be reconfigured for increased endurance of up to 100 kilocycles on the last 8 sectors of each bank.
It also supports a read prefetch unit, that increases the efficiency of Cortex M33 C-AHB bus.
Finally, STM32H5 implements a product state life-cycle scheme, including support for regression and debug of closed device upon correct authentication.

FLASH endurance

10 Kcycles endurance on all Flash memory

100 Kcycles on configurable EDATA areas, one per bank

Up to 8 Flash page can be EDATA area

EDATA area page is 6KB to accommodate different word size (16-bit + 6-bit SEC-DED)

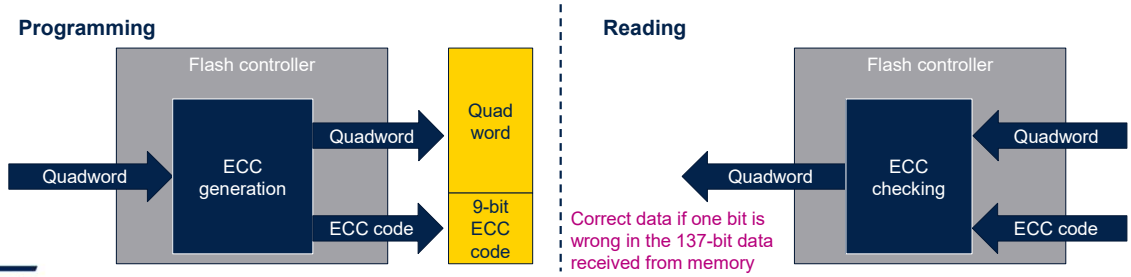
life.augmented

3

Each program / erase operation can degrade the Flash memory cell.
After an accumulation of program / erase cycles, memory cells can become non-functional, causing memory errors.
Endurance is an estimated maximum number of erase/programming sequences that the Flash memory can support without affecting its reliability.
Up to 8 sectors per bank may be reconfigured for an increased endurance of 100 kcycles, that can be used for data storage that usually needs more intensive cycling capability than code storage.
These EDATA sectors store less data to make up for the necessary redundancy for error correction.

FLASH ECC

- 9 ECC bits added to the 128-bits data line, 6bits on 16-bit data word (OTP, EDATA)
- ECC mechanism supports:
  - One error detection and correction, with optional interrupt
  - Two errors detection, with NMI generation



life.augmented

4

Data in Flash memory are 137-bits wide: nine bits are added per each quad word of 128 bits.
The ECC mechanism supports: one bit error detection and correction and two-bit error detection
When one error is detected and corrected, the ECC Correction flag is set in the Flash ECC register. An interrupt can be generated.
When two errors are detected, the ECC Detection flag is set in the Flash ECC register.
In this case, an NMI is generated. The address and bank number at which the error has been detected are captured in status registers for further investigation.
In case of error in the 16-bit word in EDATA and OTP sections, it is possible to also examine the uncorrected data in dedicated register.

FLASH read access latency

Wait states (latency)	HCLK max (MHz)				Programming delay (WRHIGHFREQ)
	VOS0 Range	VOS1 Range	VOS2 Range	VOS3 Range	
0 WS (1 cycle)	≤ 42	≤ 34	≤ 30	≤ 20	00
1 WS (2 cycles)	≤ 84	≤ 68	≤ 60	≤ 40	
2 WS (3 cycles)	≤ 126	≤ 102	≤ 90	≤ 60	01
3 WS (4 cycles)	≤ 168	≤ 136	≤ 120	≤ 80	
4 WS (5 cycles)	≤ 210	≤ 170	≤ 150	≤ 100	10
5 WS (6 cycles)	≤ 250	≤ 200	-	-	

life.augmented

5

To correctly read data from the Flash memory, the number of wait states (latency) must be correctly programmed according to the frequency of the CPU clock and the internal voltage range.
The table shows the correspondence between wait states and CPU clock frequency.
Programming delay is also needed for higher clock speed, but regardless of core voltage.

FLASH prefetch

- The Cortex-M33 fetches instructions and literal pools (constants/data) over the C-Bus and through the I-Cache
- Prefetch increases C-Bus accesses efficiency when I-Cache enabled reducing the cache refill latency
  - Prefetch is efficient in case of sequential code:
    - Allows the next sequential instruction line to be read from the Flash memory while the current instruction line is being filled in instruction cache and executed by the CPU
- Prefetch tends to increase the code execution performance at the cost of extra Flash memory accesses
- Enabling prefetch is recommended for power efficiency

life.augmented

6

The Cortex-M33 fetches instructions and literal pool constants over the C-Bus and through the instruction cache if it is enabled.
The prefetch block increases the efficiency of C-Bus accesses when the instruction cache is enabled by reducing the cache refill latency.
Prefetch is efficient in the case of sequential code; prefetch in the Flash memory allows the next sequential instruction line to be read from the Flash memory while the current instruction line is being filled in instruction cache and executed by the CPU.
Prefetch is enabled by setting the PRFTEN bit in the FLASH access control register.
It must be set only if at least one wait state is needed to

access the Flash memory.
Note that Prefetch tends to increase the code execution performance at the cost of extra Flash memory accesses, reaching over 4 coremark point per megahertz.
As Coremark code is entirely in icache (no cache miss after the first iteration), the prefetch has no impact on the Coremark score when icache is enabled.
Power efficiency is better when prefetch is enabled.

Memory erase and program operation

- Program and erase operations supported in all voltage ranges
- Page erase (8 KB), bank erase or mass erase support
- Standard programming mode: quad-word programming (4 x 32-bit data)
  - ECC is automatically calculated and added to program 137-bit line
  - Normally programming starts automatically when the 128-bit write buffer is filled
  - When writing smaller amount of data, programming can be forced
- OTP, EDATA programming is possible by 16-bit or 32-bit word only
- Operation status register to be able to recover is case of system reset during programming or erasing operation

Parameter	Typical value
TTprog (time to program) 137-bit flash line or 16-bit OTP/EDATA	31 µs
Tmass_erase (2 banks)	2 s
Terase (one sector)	2 ms

life.augmented

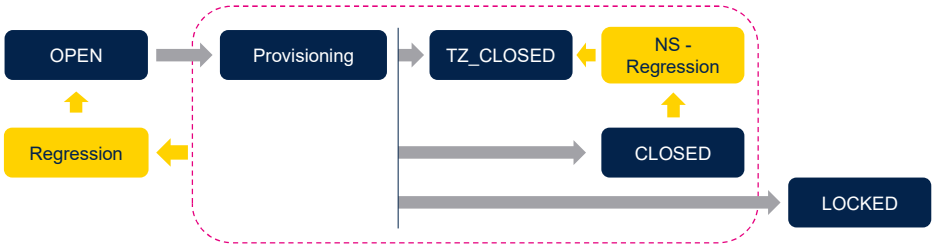
7

Read, program and erase operations are supported in all voltage ranges.
When trustzone is enabled, the programing and erase must also respect the isolation boundaries.
Erase can be performed either with a sector granularity, or for one bank or both banks.
In the latter case, this is called a mass-erase.
The ECC code is calculated and added to the data, so that 137 bits are actually programmed.
The contents of the Flash memory currently being modified are not guaranteed if a reset occurs during a Flash memory program or erase operation.
The status of the Flash memory can be recovered from the FLASH operation status register when a system reset

occurs during a Flash memory program or erase operation.
It is the software’s responsibility to check the Flash memory status and to take corrective actions accordingly.

Life cycle management

- An RDP replacement
  - A natural progression from virgin to closed state with no debug access
  - Regression possible only under strict conditions



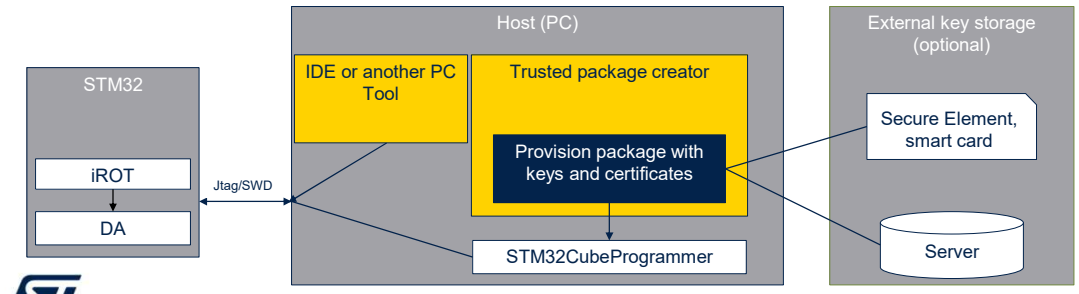
life.augmented

8

The figure shows a simplified scheme of product state progression, grey arrows represent increasing protection during manufacturing or development, while orange arrows represent ways of regressing to a previous state upon successful authorization.
The process of assigning the authorization keys as well as the extent of permitted regression is called provisioning.
It is also possible to exit the cycle of progression and regression permanently by assigning locked state.
Progressing to more closed state is the normal product life cycle, that does not require security measures. Transition in direction to open state is a regression, controlled by the debug authentication control.
If debug unlock policy is set to “locked”, no regression is accepted.

STM32H5 controlled regression

- Reopening debug
  - Provisioning of the keys in the device (provisioning product state)
  - PC tools (All tools using JTAG/SWD must be updated to version with STM32H5 support)
    - Tools may regress the device to Open state
    - Tools may reopen the debugging for one session



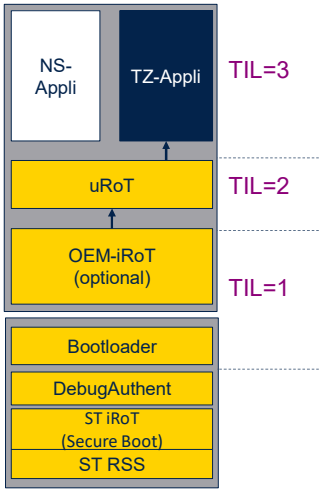
life.augmented

9

Tools are provided to facilitate the provisioning, integrate it with Secure Firmware Install (SFI) scheme and distribute certificates allowing selected developers to establish temporary debug sessions.
The regression and debug rights may also be delegated.

Temporal isolation protection

- HDP is controlled by TIL, used for Arm PSA
  - In STM32H5 independent on secure area (may or may not overlap)
  - Intended to protect assets that are only needed in startup phase
  - TIL level is controlled by SBS
  - Based on TIL, some assets are blocked
  - TIL levels have associated slots in key storage (detailed later)
    - Separate TIL signal for OBK is called OBK-TIL



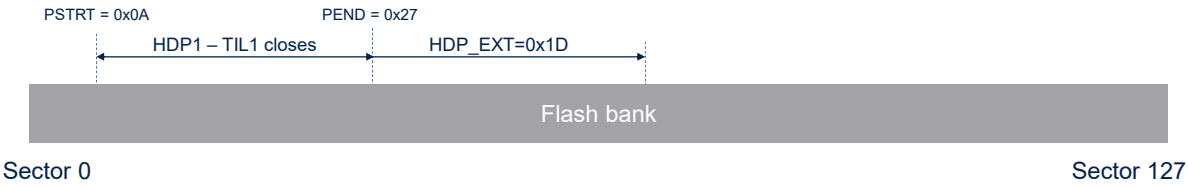
life.augmented

10

Secure Hide Protection (HDP) is controlled by Temporal Isolation Level (TIL), used for Arm Platform Security Architecture (PSA)
Three levels are available to facilitate secure boot process stages isolation.
Each level increase blocks access to the lower levels, which only become available again on next reboot.
Same mechanism works on smaller scale in dedicated key storage mechanism called the OBK.

HDP control – User Flash

- Each bank has independent controls
- TIL1 HDP area is programmed in User OB by start and end sector
- HDP are can be extended with several sectors using FLASH_HDP_EXT
  - Register value starts as 0 on reset
  - New written value must always be same or larger than previous
  - Extension is set by TIL=2 code and prevents access from TIL=3



life.augmented

11

The area associated with TIL 1, which is blocked by incrementing to level 2, is configured using option bytes. The threshold between TIL 2 and TIL 3 is a volatile setting in flash register, called the HDP extension.
If the two banks are swapped, the protection defined to physical Bank1 remains on the physical Bank1, unaffected by swapping

FLASH TrustZone support

- TrustZone activated by option byte
- Area based watermark
  - One area per bank
  - Stored in Option Byte, the area is secure since Option Byte load phase
  - Marked by start (SECWM_PSTRTx) and end sector (SECMW_PENDx)
- Bitmap based
  - Same principle as with privilege, each sector can be set individually
  - FLASH_SECBBn_p registers, 4 for each bank
  - Volatile setting only (until reset)

life.augmented

12

With exception of the STM32H503 the TrustZone isolation is available on the STM32H5 series
When TrustZone security is active, a part of the Flash memory can be protected against non-secure read and write accesses.
Deactivation of TrustZone is only possible in the open product state or by orderly regression.
Up to two different non-volatile secure areas can be defined by option bytes and can be read or written only by a secure access : one area per bank with a sector granularity.
Watermark secure areas are complemented by bitmap-based definition allowing for non-continuous secure areas.

Privilege flash access protection

- Each sector can be set individually as privileged
- The setting is volatile only – not stored in OB configuration
  - Four FLASH_PRIVBB1_p registers for bank1
  - Four FLASH_PRIVBB2_p registers for bank2
- 256 bits total, each represents one sector

Secure + privilege	Non-Secure + privilege
Secure + Non-privilege	Non-Secure + Non-privilege

life.augmented

13

In each security domain, the privilege level of each flash sector is programmable: either unprivileged or privileged, by means of dedicated registers. 4 quadrants of isolated worlds are thus obtained:

- Secure privilege
- Secure non-privilege
- Non-secure privilege
- Non-secure non-privilege

Isolation may also be applied on the sectors of the high cycle area.

OB Key storage area

- Option Byte Key storage area is a dedicated storage area with tight access conditions
  - Privilege and secure access only (except when TZ not active)
  - Correct TIL is required to access the keys
  - Storage is invalidated on regression using EPOCH counter

TIL	From offset	To offset
reserved	0x0000	0x00FF
1	0x0100	0x08FF
2	0x0900	0x0BFF
3	0x0C00	0x17FF
3 non secure	0x1800	0x1FEF

Still only accessible by secure code

life.augmented

14

Option Byte Key stotage (OBK) is a mechanism dedicated to storage of small data with high protection requirements, mostly cryptographic keys.
The keys stored in this area are protected not only by the TIL, but also encrypted using EPOCH counter in the initialization vector.
The EPOCH counters are incremented on each regression, making it impossible to recover the OBK contents after regression was made.
New provisioning or installation of keys must be made.
Secure keys placed in OBKey sectors, and protected by 9 bits of ECC.
In case of tamper, all keys are read as 0x00s.

OB key controls

- To allow for modifications, two sectors are used to store the keys
  - In modification, unmodified part is copied
- Keys are controlled by FLASH_SEC_OBKCR register providing services:
  - Swap (copy contents to alternate sector and make it current)
  - Erase alternate sector
  - Write
  - Peek alternate sector
  - Alternate NS register is only available when TZ_STATE=0xC3 (disabled)
- Keys are supposed to start at addresses divisible by 16 (0x10, 128bit)
- Larger key occupies several 16 bytes “slots”

life.augmented

The OBK uses two dedicated flash sectors, one is used as working while the other stages the modifications.
When the contents update is completed, they switch places.
The keys stored may serve purposes of attestation, encryption, authentication, secure storage or storage of important hash values.

Thank you

life.augmented

In addition to this presentation, you can refer to the following presentations:

- Security-Enhanced key storage
- STM32H5-Security-Lifecycle
- STM32H5-Security-Enhanced anti-tamper.