Hello, and welcome to this presentation of the STM32F7 hash processor.
Hash peripheral is in charge of efficient computing of message digest. A digest is a fixed-length value computed from an input message. A digest is unique - it is virtually impossible to find two messages with the same digest. The original message cannot be retrieved from its digest. Hash digests and Hash-based Message Authentication Code (HMAC) are widely used in communication since they are used to guarantee the integrity and authentication of a transfer.
The hash processor supports widely used hash functions including Message Digest 5 (MD5), Secure Hash Algorithm SHA-1 and the more recent SHA-2 with its 224- and 256-bit digest length versions.
A hash can also be generated with a secret-key to produce a message authentication code (MAC).
The processor supports bit, byte and half-word swapping. It supports also automatic padding of input data for block alignment.
The processor can be used in conjunction with the DMA for automatic processor feeding.
All supported hash functions work on 512-bit blocks of data. The input message is split as many times as needed to feed the hash processor. Subsequent blocks are computed sequentially. MD5 is the less robust function with only a 128-bit digest. The SHA standard has two versions SHA-1 and the more recent SHA-2 with its 224- and 256-bit digest length versions.
The hash-based message authentication code (HMAC) is used to authenticate messages and verify their integrity. The HMAC function consists of two nested hash operations:

- HMAC(message) = Hash([[key | pad) ⊕ 0x5C) | Hash([[key | pad) ⊕ 0x36) | message]]
- Hash function is any of the ones supported by the peripheral

The hash function involved in the HMAC computation can be any one supported by the peripheral: MD5, SHA-1 or SHA-2
The hash processor complies with the international standards for Secure Hash Algorithms (SHA), Message Digest algorithms (MD5) and for Message Authentication Code (MAC).
This simplified block diagram of the hash processor shows the basic data flow and control modules. The hash processor processes 512-bit data blocks and generates digests of up to 256 bits depending on the algorithm. Input data may be swapped before entering the core unit where they will be processed to generate a simple hash or a message authentication code (MAC).
An interrupt in the nested vectored interrupt controller (NVIC) is triggered when a hash digest has been successfully calculated or when the hash processor is ready to accept a new block of data.

In Direct memory access (DMA) mode, requests are generated internally for incoming data. The DMA channel must be configured in Memory-to-peripheral mode with a data size equal to 512 bits.
These are the times it takes top process a single block of data depending on the chosen algorithms. HCLK is the CPU clock and can go as high as 216MHz. Note that main benefit of using a hardware accelerator is to increase speed and save power compared to a full software implementation of the hash functions.
Here is an overview of the status of the hash processor in each of the low-power modes. Hash operations are not possible when the device is in Stop mode.

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run</td>
<td>Active.</td>
</tr>
<tr>
<td>Sleep</td>
<td>Active. Peripheral interrupts cause the device to exit Sleep mode.</td>
</tr>
<tr>
<td>Stop</td>
<td>Frozen. Peripheral registers content is kept.</td>
</tr>
<tr>
<td>Standby</td>
<td>Powered-down. The peripheral must be reinitialized after exiting Standby mode.</td>
</tr>
</tbody>
</table>
This is a list of peripherals related to the hash processor. Please refer to CRYP peripheral trainings if you want to know more about cryptographic engines. Refer to training on the DMA peripheral for information on how to configure the hash channel.
For more details, please refer to this user manual available on our website.

- For more details and additional information, refer to following:
  - UM0586: STM32 Cryptographic Library