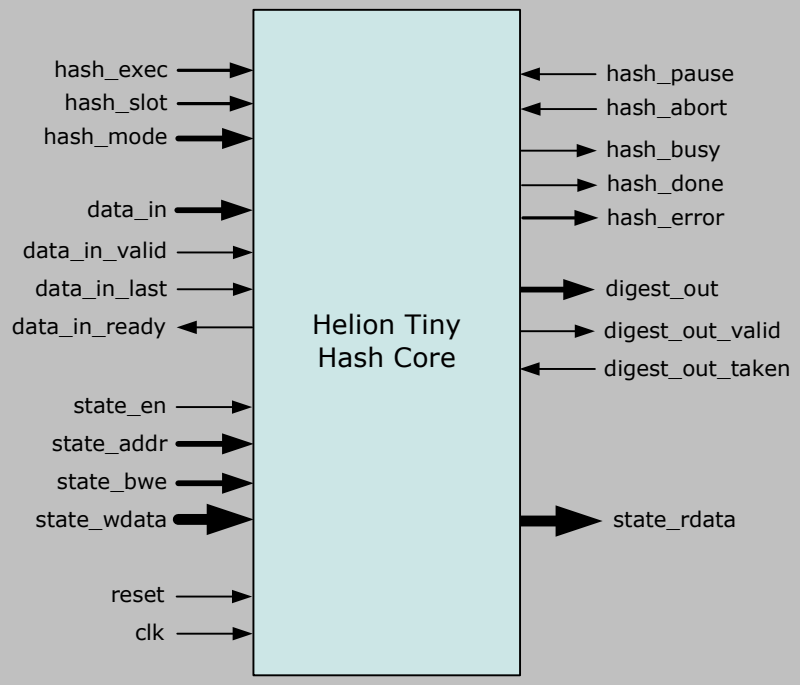
Helion Technology

PRODUCT BRIEF | **Tiny Hash IP cores for ASIC**



Features

- Implements one or more of SHA-1, SHA-224, SHA-256, SHA-384 & SHA-512 secure hash algorithms defined in FIPS PUB 180-3
- Supports Keyed Hashing for Message Authentication (HMAC) to FIPS 198-1
- Performs full message padding according to FIPS PUB 180-3
- Provides high functionality for low resource, low data rate applications
- Runs up to four concurrent hashes each using different hash algorithms
- Supports full state unload/reload to optimise hashing of interleaved data
- All state stored in small single-ported RAM or Register file for efficiency
- Optimised for use in ASIC

Deliverables

- Fully synthesisable RTL source code
- VHDL/Verilog testbench with test vectors
- User documentation

Overview

The Helion Tiny Hash Core family for ASIC offers a combination of high functionality and low resource usage for lower data rate applications. The core is available in versions which support any combination of the secure hashing algorithms described in the Secure Hash Standard, FIPS PUB 180-3; namely SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512. It can also support the standard Hash-based Message Authentication Code (HMAC) algorithm described in FIPS PUB 198-1 which is widely used for data authentication and integrity checking in a number of common data security protocols.

As standard the core supports up to four concurrent hash calculations (or two HMAC calculations), and offers full core state unload and reload to greatly improve system throughput when processing interleaved or packet-based data streams. Simple synchronous interfaces ensure easy system integration whether employed as a hashing accelerator for an embedded processor, or connected directly into a datapath.

Helion Technology Limited
Ash House, Breckenwood Road,
Fulbourn, Cambridge CB21 5DQ, England

Functional Description

Background
The Helion Tiny Hash core for ASIC is fully configurable and can implement any combination of the NIST secure hashing algorithms specified in FIPS PUB 180-3; namely SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512. It can also support the newer SHA-512-224 and SHA-512-256 hash algorithms defined in DRAFT FIPS PUB 180-4.

Multi channel hashing
The core is able to switch hash algorithm dynamically between message blocks, allowing any of the hash algorithms supported by the core to be selected for use on a message block by block basis. As standard, the core internally provides four message slots allowing support for four concurrent hash calculations, each using a different hash algorithm if required. This enables up to four message streams to be hashed simultaneously. This capability may be extended using the optional state unload and reload interface, where the core can very efficiently handle more than four interleaved message streams using external state storage.

Core Operation
The user initiates a new hash operation by applying a "RUN" operation code to the 2-bit wide *hash_exec* input. During the same clock cycle, the *hash_mode* and *hash_slot* inputs are used to indicate the required hash algorithm (with or without HMAC) and which of the four internal memory slots is to be used for the operation. The core then asserts the *hash_busy* flag to show that an operation is underway, and indicates its readiness to accept message data on the byte-wide *data_in* port, via simple *data_in_valid* and *data_in_ready* handshake signalling. The message for hashing is then pushed into the core by the user.

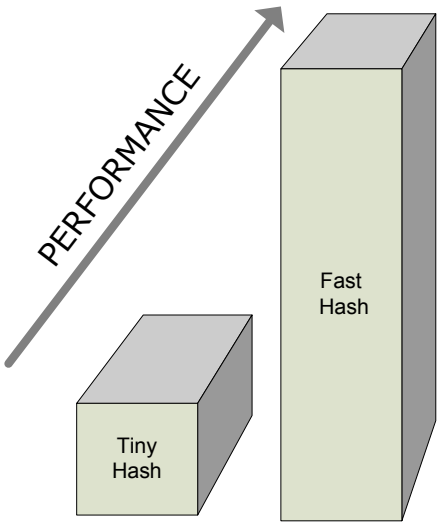
End of Message
At the end of the message, the user flags the last message byte by asserting the *data_in_last* marker input as the last byte is transferred into the core. This tells the core to append message padding in accordance with FIPS PUB 180-3.

Once the core has completed hashing of the final message block it deasserts the *hash_busy* flag and strobes the *hash_done* output for a clock cycle. At the same time, the validity of the message digest output is indicated by the core asserting *digest_out_valid*. The user may then read the message digest from the byte-wide *digest_out* port using the *digest_out_taken* handshake signal. Once the full message digest has been transferred, the core deasserts *digest_out_valid*, and the core is then ready to start work on a new message.

Other Controls
Two further control inputs are provided to allow suspension of the current hash operation; *hash_pause* stops the current core operation at the end of the present message block for state unload and reload operation; and *hash_abort* halts the current operation immediately and returns the core to its idle state. A new hash operation may then be requested in the very next clock cycle.

Other Operations
As mentioned above, the *hash_exec* input is used to initiate a new operation. This tells the core to either run a new message hash, resume a previous one where the state has been unloaded and reloaded, or to input a new HMAC key. In the latter case, the HMAC key is loaded via the normal data input path, whereas state information is unloaded and reloaded via a dedicated memory-mapped RAM interface to allow access to the core's internal state whenever it is not busy. A *hash_error* output indicates when an operation has been requested by the user that is outside the capability of the core, e.g. an unsupported hash algorithm has been selected.

Core Choice



The **Tiny** Hash core is one of two hashing solutions available from Helion, and is aimed at lower data rate applications up to a few tens of Mbps, offering low resource utilisation together with a rich feature set.

The Tiny Hash core family provides a number of core versions, each sharing a common user interface whilst providing support for one or more hashing algorithms and optionally including HMAC processing.

The core design is fully parameterised in terms of its algorithmic support, so for applications not needing access to all the available algorithms, significant logic resources may be saved.

Where more throughput is required, Helion also offer a **Fast** Hashing core family, which is covered in a separate datasheet. Please see http://heliontech.com/fast_hash.htm for more details on this solution.

Core Throughput

The tables below show the number of cycles and the maximum data throughput as a function of core clock frequency, for each of the algorithms running in the Tiny Hash core.

algorithm	SHA-1	SHA-224/256	SHA-384/512
size of hash block	512-bits	512-bits	1024-bits
clock cycles used per hash block	1013	1013	1317
data throughput (Mbps per MHz)	0.505	0.505	0.777

For any specific application, the above figures may be used with an appropriate and achievable core clock frequency to determine actual message processing time. Note that for HMAC there are additional per message overheads associated with the HMAC algorithm which can be significant for shorter message lengths. In this case additional performance margin may be required to achieve your desired HMAC throughput - please contact Helion to discuss your HMAC requirements in more detail.

Logic Utilisation and Performance

The table below shows typical area and maximum clock rates for popular configurations of the Helion Tiny Hash core.

	Helion Tiny Hash cores			
version	**SHA-1 only**	**SHA-256 only**	**SHA-512 only**	**All algorithms**
typical gatecount + SP RAM	<6k gates 32x32 RAM	<8k gates 40x32 RAM	<13k gates 40x64 RAM	<17k gates 40x64 RAM
typical max clock rate (65nm)	300MHz	300MHz	300MHz	300MHz

The core has been designed to use the absolute minimum area for each user configuration and very efficiently makes uses of a small single-port RAM or register file to implement the significant amount of internal state storage required by all secure hash algorithms. This results in a solution that is much more compact and power efficient when compared to more traditional designs where this state might usually be stored in discrete registers.

Note that exact figures will depend significantly on the target library used, as well as the synthesis method and options, so these numbers should be treated as preliminary guidance only.

Looking for Higher Rates?

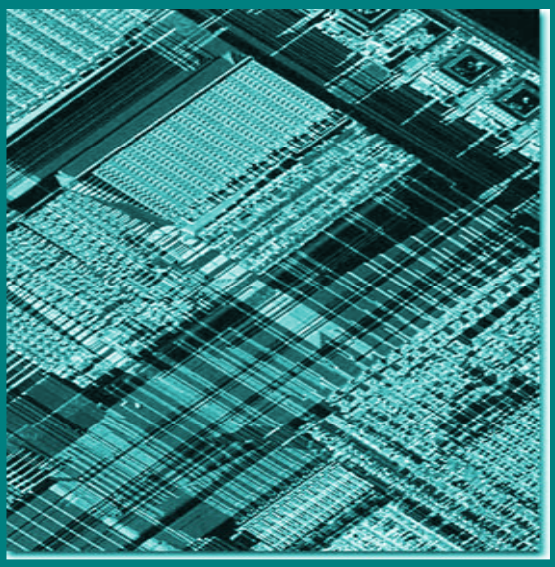
The Helion Tiny Hash core is designed specifically for lower data rate applications, where its multi-algorithm capability and optional built-in HMAC support might be valuable. However where data throughputs exceed the low 100's of Mbps capability of this design, a higher rate solution may be more appropriate.

Helion offer a Fast Hash core family which supports single and pairs of hash algorithms at low Gbps rates. Please take a look at our Fast Hash webpage at http://www.heliontech.com/fast_hash.htm, or contact Helion for more information on faster hashing core solutions.

Ordering Information

Before ordering, the first thing to decide is which algorithms you would like to support - one or more from this list; SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512. If you require support for HMAC this should also be specified at time of order. Finally, the number of memory slots required (between 1 and 4) should be indicated.

If some of these choices are unclear, or you would just like to go over the options available, we are always happy to discuss the alternatives and help select the best solution for your application.



"I would say that the support from Helion has been outstanding from my initial contact all the way through implementation. I was very impressed when I asked questions on weekends and received answers the same day.
The core was actually so easy to use I completed my implementation almost a month early.
The product performed exactly as advertised and the price made it a very affordable option to add to our system."

Jim Cassey
Sr. Hardware Engineer
Delta Digital Video

About Helion

Founded in 1992, Helion is a long established British company based in Cambridge, England. We offer a range of product-proven Data Security and Lossless Compression IP cores, backed by a team of highly experienced engineers, proudly developing and supporting a world-class portfolio.

Our aim is to offer our customers…

Innovation
Helion works hard to anticipate, understand and then deliver great solutions for its customers. As an example, Helion offered the world's first commercial AES core back in 2001, even before the industry had fully adopted the algorithm. This process continues unabated today, with new products in development that will lead the field.

High Performance
Helion IP is specially designed and optimised for each target technology. This means lots of work for us, but this approach yields amazing results for our customers. We always aim for the best in class performance and lowest utilisation in any given ASIC or FPGA target.

High Quality
IP should be problem free, so we always go the extra mile to ensure a smooth and trouble free integration phase for our products. We realise that our customers are putting their faith in us, and want to repay that with an outstandingly easy deployment.

Ease of Use
Helion engineers have many years of real product development experience, and so our IP is designed to be used in realistic situations. It is flexible and well thought through - the result being that it is simple to drop into your system.

See how we achieve all this by visiting our Clients page at http://www.heliontech.com/clients.htm

More Information

For more detailed information on this or any of our other products and services, please contact Helion and we will be pleased to discuss how we can assist with your individual requirements.

Helion Technology Limited
Ash House, Breckenwood Road,
Fulbourn, Cambridge CB21 5DQ, England

tel: +44 (0)1223 500 924 email: info@heliontech.com
fax: +44 (0)1223 500 923 web: www.heliontech.com