



SHA-384, SHA-512 Hashing, Fast (Helion)

May 15, 2007

Product Specification



Helion Technology Limited

Ash House, Breckenwood Road,
Fulbourn, Cambridge CB1 5DQ, England
Phone: +44 1223 500 924
Fax: +44 1223 500 923
E-mail: helioncores@heliontech.com
URL: www.heliontech.com

Features

- Available under terms of the SignOnce IP License
- Supports both SHA-384 and SHA-512 hash algorithms as specified by FIPS PUB 180-2
- Provides twice the data throughput of the SHA-1 hash algorithm
- Designed specifically for high throughput, multi-Gigabit/second applications
- Performs automatic message length calculation and padding insertion
- Optional user initialisation of IVs for efficient HMAC support
- Optional mid-message hash state unload/reload feature for handling fragmented messages
- Highly optimised for use in all Xilinx FPGA technologies

AllianceCORE™ Facts	
Provided with Core	
Documentation	User Guide
Design File Formats	Xilinx netlist; VHDL or Verilog source code also available
Constraints Files	.ucf
Verification	VHDL or Verilog test bench with NIST FIPS test vectors; VHDL or Verilog Simulation models
Instantiation templates	VHDL, Verilog
Reference designs & application notes	Detailed HMAC application notes; HMAC wrapper also available
Additional Items	Example ModelSim scripts
Simulation Tool Used	
ModelSim PE 6.1e	
Support	
Support provided by Helion Technology Limited	

Table 1: Example Implementation Statistics – Dual SHA-384 and SHA-512 mode

Family	Example Device	Fmax ² (MHz)	Slices	IOB ¹	GCLK	BRAM	MULT	DCM/DLL	MGT	PPC	Design Tools
Spartan-3™	XC3S2000-5	82	1497	587	1	1	0	0	N/A	N/A	ISE 9.1.02i
Virtex-4™	XC4VLX25-11	133	1487	587	1	1	0	0	0	0	ISE 9.1.02i
Virtex-5™	XC5VLX30-3	188	606	587	1	0	0	0	0	0	ISE 9.1.02i

Notes:

- 1) Assuming all core I/Os and clocks are routed off-chip
- 2) Fmax is quoted assuming all core inputs are sourced from flip-flops, and all core outputs drive flip-flops; this has been done to best represent real applications.

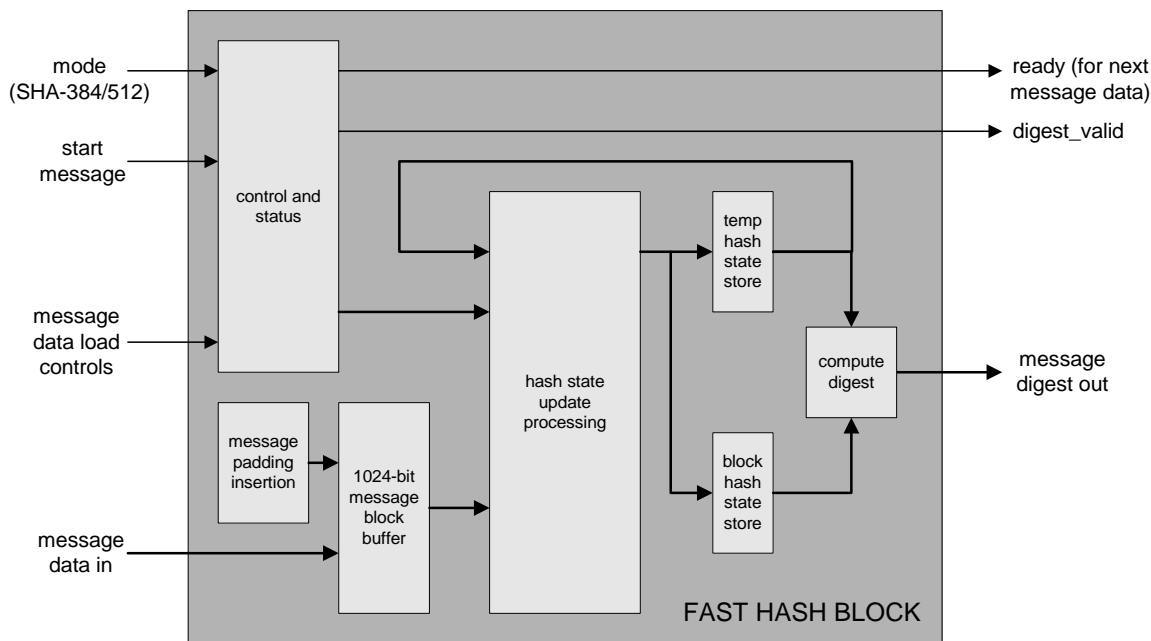


Figure 1: Fast SHA-384/512 Hashing Core Block Diagram

Table 2: Example Implementation Statistics – SHA-512 only mode

Family	Example Device	Fmax ² (MHz)	Slices	IOB ¹	GCLK	BRAM	MULT	DCM/DLL	MGT	PPC	Design Tools
Spartan-3™	XC3S2000-5	80	1369	586	1	1	0	0	N/A	N/A	ISE 9.1.02i
Virtex-4™	XC4VLX25-11	135	1381	586	1	1	0	0	0	0	ISE 9.1.02i
Virtex-5™	XC5VLX30-3	188	605	586	1	0	0	0	0	0	ISE 9.1.02i

Table 3: Example Implementation Statistics – SHA-384 only mode

Family	Example Device	Fmax ² (MHz)	Slices	IOB ¹	GCLK	BRAM	MULT	DCM/DLL	MGT	PPC	Design Tools
Spartan-3™	XC3S2000-5	83	1369	458	1	1	0	0	N/A	N/A	ISE 9.1.02i
Virtex-4™	XC4VLX25-11	135	1381	458	1	1	0	0	0	0	ISE 9.1.02i
Virtex-5™	XC5VLX30-3	183	608	458	1	0	0	0	0	0	ISE 9.1.02i

Notes:

- 1) Assuming all core I/Os and clocks are routed off-chip
- 2) Fmax is quoted assuming all core inputs are sourced from flip-flops, and all core outputs drive flip-flops; this has been done to best represent real applications.

Applications

The SHA-384 and SHA-512 secure hash algorithms are typically used in message authentication and digital signature applications, where a message or file requires tamper protection or origin authentication. They are approved for use in US Federal applications for securing sensitive unclassified information. The SHA-384 algorithm is also a requirement of the US National Security Agency “Suite B” set of cryptographic algorithms for the protection of top secret information.

Both SHA-384 and SHA-512 provide improved security and higher data throughput per MHz than the widely used SHA-1 algorithm. Specific applications might include hardware implementations of the Digital Signature Algorithm (DSA), where a hash function is used to generate and verify signatures for data integrity and origin authentication (specified in FIPS PUB 186).

General Description

SHA-1 (“SHA” standing for “Secure Hash Algorithm”) was developed by the National Institute of Standards and Technology (NIST) as a replacement for its earlier SHA algorithm, and was re-published as part of a Federal Information Processing Standard (the latest version being FIPS Publication 180-2). In FIPS Publication 180-2 a new family of secure hash algorithms commonly known as SHA-2 were introduced which consist of three different algorithms; SHA-256 which in common with SHA-1 is a 32-bit algorithm that operates on 512-bit message data blocks but with improved data security and higher throughput, and SHA-384 and SHA-512 which are 64-bit algorithms which operate on 1024-bit message data blocks and provide even higher data security and throughput.

The basic function of each hash algorithm is to produce a compressed “digest” of a longer message or file. This digest is broadly equivalent to a conventional checksum, but has a number of special cryptographic properties; the important ones being that the same digest should never be generated by two differing messages, and that it should not be possible to re-generate any of the original message from the digest. These properties make the hashed message digest very useful for proving that a message has not been changed in transit. A digest appended to the message before transmission (usually before the whole message has been encrypted for security) can be compared to a locally computed version on receipt. If they are the same, you can conclude with some degree of certainty that the message has not been altered since the transmitted hash was generated. Taking this a stage further, the hash can be uniquely “keyed” by using a common technique called HMAC (described in RFC2104), so that the originator of the digest can be authenticated too.

The Helion Fast SHA-384/512 hashing core has been designed to implement SHA-384 and SHA-512 in hardware for high throughput applications. Where the SHA-384 algorithm only is required the mode pin may be tied to a logic ‘1’ to minimize the amount of slices used. Where the SHA-512 algorithm only is required the mode pin may be tied to a logic ‘0’ to minimize the number of slices used. The logic savings made when implementing a single mode in this way will depend on the device family used as shown in Tables 2 & 3.

Both hash algorithms handle the message data in the same way. They each process an arbitrary length input message by operating on successive 1024-bit blocks of data, to produce a final message digest of length 384-bits in the case of SHA-384, or 512-bits in the case of SHA-512. The message data is conveniently entered as a series of 64-bit words, and then at the end, the resulting digest emerges as either a 384 or 512-bit parallel value. Although the hash algorithms do subtly differ in the way they generate their digests, the Helion core has been designed to make them appear exactly the same at the user interface level.

Functional Description

The Helion Fast SHA-384/512 hashing core is extremely simple to use. Once a new message is initiated, message data can be loaded into the core as a series of 64-bit words, with the core indicating each time it is ready to accept more data. This message loading simply continues until the end of the message is reached.

Both hashing algorithms specify a padding scheme which needs to be applied to the end of every message regardless of length, and this is handled completely automatically and transparently to the user inside the core. As the word containing the final message byte is being loaded, this should be indicated to the core, together with its position within the final word (the hashing core can therefore accept a message of any arbitrary number of bytes). The core will then internally generate and append the specified padding, and complete the necessary processing to produce the final digest output. The digest will be indicated as being valid, after which a new message may be started.

Beyond this normal operation, the core optionally allows the user to load an IV (Initialisation Vector) into the hash engine at the start of a new message. This feature can be useful for some optimised HMAC implementations, where a pre-calculated IV can save some of the per-message HMAC overhead. The core can also optionally support mid-message state unload and reload; this is useful where messages are fragmented and interleaved, as they may be in some network environments. Please feel free to contact Helion Technology directly to discuss these advanced options in more detail.

Figure 1 shows the the main internal functional blocks within the basic Helion Fast Hashing core (the optional IV and state loading functionality is omitted for clarity). Incoming message data words are stored in an internal 1024-bit block buffer for subsequent processing, with end of message length and padding insertion being handled automatically and transparently by a dedicated block. Once a data block has been loaded, it is processed according to the selected algorithm by application of a sequence of complex and varying logical and arithmetic functions over a number of iterations. Temporary intermediate results are stored for each iteration, and then at the end of each block process, these are used together with the previous block state to compute the running digest. An overall control and status block keeps track of operation and handles all interface and datapath timing.

Core Data Throughput Capability

The maximum data throughput capability of the cores is directly proportional to the master clock frequency used in the implementation. For both SHA-384 and SHA-512 the maximum data throughput as a function of clock frequency is as follows;

$$\text{Max throughput (Mbps)} = (1024 / 82) \times \text{Master Clock Frequency (MHz)}$$

The highest Master Clock Frequency which may be used is both mode and technology dependent; absolute maximum frequencies for current Xilinx families in various mode configurations are shown in Tables 1, 2 and 3 at the start of this datasheet. Please contact Helion Technology directly to discuss your throughput requirements in more detail.

Core Modifications

As detailed previously, a number of options are available at time of ordering. These are as follows;

- optional user IV load capability
- optional mid-message hash state unload/reload capability

Please contact Helion Technology directly to discuss these options in more detail.

Core I/O Signals

The core signal I/O have not been fixed to specific device pins to provide flexibility for interfacing with user logic. Descriptions of all signal I/O are provided in Table 4. Note that the optional additional User IV loading and state unload/reload capabilities require additional I/O to that detailed below; please contact Helion Technology for more details.

Table 4: Core I/O Signals.

Signal	Width	Signal Direction	Description
clk	1	Input	Master Clock Input
reset	1	Input	Master asynchronous reset; 1 = reset
mode	1	Input	Hash algorithm mode (1= SHA-384, 0 = SHA-512)
start_message	1	Input	Start message control input
data_in	64	Input	Message data word input
load	1	Input	Load message data word
load_last	1	Input	Load last word of message (indicates current data word contains final message byte)
last_byte	2	Input	Last byte index of last word (indicates which byte in word is the final message byte)
ready	1	Output	Core ready to accept next message word
message_digest	512	Output	Message digest result (SHA-384 only uses most significant 384-bits)
digest_valid	1	Output	Message digest valid flag

Verification Methods

The Helion Fast Hashing cores have been thoroughly verified under simulation using a combination of NIST FIPS and RFC test vectors; these tests are also supplied as part of the core deliverables to demonstrate operation. In addition, the cores have been carefully tested in real Xilinx devices, and have all been successfully used many times over in real products shipping in volume.

Recommended Design Experience

Users should be familiar with HDL methodology and Xilinx design flows including VHDL/Verilog component instantiation, synthesis, implementation and simulation.

Ordering Information

This product is available directly from Xilinx AllianceCORE member Helion Technology Limited under the terms of the SignOnce IP License. Please contact Helion Technology for pricing and additional information about this product. Contact information for them is on the front page of this datasheet. To learn more about the SignOnce IP License program, contact Helion Technology or visit the web:

Email: commonlicense@xilinx.com
URL: www.xilinx.com/ipcenter/signonce

Related Information

Hash algorithm specifications

For more detailed information on SHA-384 and SHA-512, the latest specification (NIST FIPS Publication 180-2) may be downloaded from the NIST website at;

<http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf>.

Xilinx Programmable Logic

For information on Xilinx programmable logic or development system software, contact your local Xilinx sales office, or:

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: +1 408-559-7778
Fax: +1 408-559-7114
URL: www.xilinx.com