



Modular Exponentiation Engine for RSA and DH (ModExp)

February 16, 2007

Product Specification



Helion Technology Limited

Ash House, Breckenwood Road,
Fulbourn, Cambridge CB21 5DQ, England
Phone: +44 1223 500 924
Fax: +44 1223 500 923
E-mail: helioncores@heliontech.com
URL: www.heliontech.com

AllianceCORE™ Facts	
Provided with Core	
Documentation	User Guide
Design File Formats	Xilinx netlist
Constraints Files	.ucf
Verification	VHDL or Verilog test bench; VHDL or Verilog Simulation models
Instantiation Templates	VHDL , Verilog
Reference Designs & Application Notes	
Additional Items	Example ModelSim scripts
Simulation Tool Used	
ModelSim PE 6.1e	
Support	
Support provided by Helion Technology Limited	

Features

- Implements $Z = Y^E \text{ mod } M$ Modular Exponentiation function used in Public-Key Cryptography
- Ideal for hardware acceleration of RSA, Diffie-Hellman and DSA (FIPS 186-3) algorithms
- Simple 32-bit RAM interface
- Supports 768, 1024, 1536 and 2048-bit operand lengths
- Also supports short Exponent lengths at higher performance e.g. 180-bit for Diffie-Hellman
- Available in a choice of versions allowing user to trade-off area and performance
- Capable of up to ~40 1024-bit RSA operations per second (E=1024, M=1024)
- Capable of up to ~220 1024-bit Diffie-Hellman operations per second (E=1024, M=180)
- Special reduced resource versions available supporting Diffie-Hellman Oakley Group 1, 2, 14 or 15 for Internet Key Exchange (IKE) applications
- Available under terms of the SignOnce IP License

Table 1: Example Implementation Statistics – TINY32 ModExp Core

Family	Example Device	Fmax ¹ (MHz)	Slices	IOB ²	GCLK	BRAM	MULT/ DSP48	DCM / CMT	MGT	PPC	Design Tools
Spartan-3™	XC3S1500-5	97	272	81	1	3	0	0	N/A	N/A	ISE 9.1.01i
Virtex-4™	XC4VLX25-11	138	272	81	1	3	0	0	N/A	N/A	ISE 9.1.01i
Virtex-5™	XC5VLX30-3	253	159	81	1	1	0	0	N/A	N/A	ISE 9.1.01i

Notes:

1) 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.

2) Assuming all core I/Os and clocks are routed off-chip

Modular Exponentiator (ModExp)

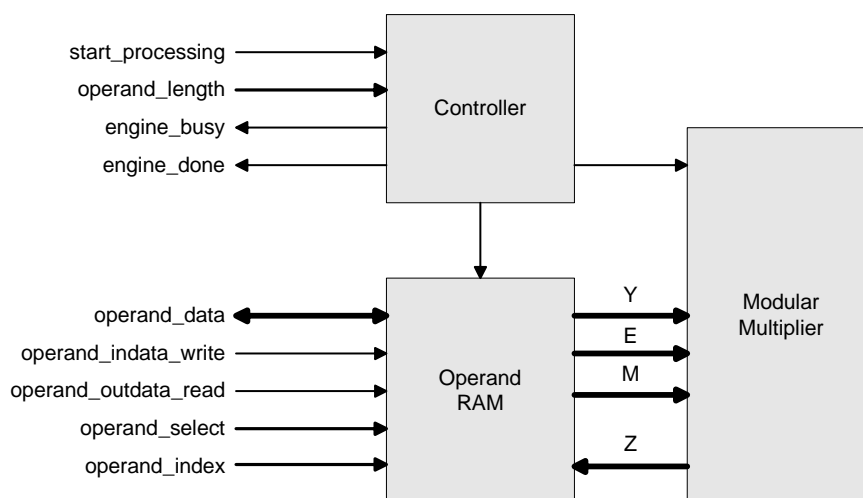


Figure 1: Helion Modular Exponentiator Block Diagram

Table 2: Example Implementation Statistics – STD64 ModExp Core

Family	Example Device	Fmax ¹ (MHz)	Slices	IOB ²	GCLK	BRAM	MULT/ DSP48	DCM / CMT	MGT	PPC	Design Tools
Spartan-3™	XC3S1500-5	144	797	81	1	1	0	0	N/A	N/A	ISE 9.1.01i
Virtex-4™	XC4VLX25-11	244	797	81	1	1	0	0	N/A	N/A	ISE 9.1.01i
Virtex-5™	XC5VLX30-3	277	267	81	1	1	0	0	N/A	N/A	ISE 9.1.01i

Table 3: Example Implementation Statistics – STD128 ModExp Core

Family	Example Device	Fmax ¹ (MHz)	Slices	IOB ²	GCLK	BRAM	MULT/ DSP48	DCM / CMT	MGT	PPC	Design Tools
Spartan-3™	XC3S1500-5	139	1052	81	1	1	0	0	N/A	N/A	ISE 9.1.01i
Virtex-4™	XC4VLX25-11	230	1052	81	1	1	0	0	N/A	N/A	ISE 9.1.01i
Virtex-5™	XC5VLX30-3	266	392	81	1	1	0	0	N/A	N/A	ISE 9.1.01i

Table 4: Example Implementation Statistics – STD256 ModExp Core

Family	Example Device	Fmax ¹ (MHz)	Slices	IOB ²	GCLK	BRAM	MULT/ DSP48	DCM / CMT	MGT	PPC	Design Tools
Spartan-3™	XC3S1500-5	121	1826	81	1	1	0	0	N/A	N/A	ISE 9.1.01i
Virtex-4™	XC4VLX25-11	210	1826	81	1	1	0	0	N/A	N/A	ISE 9.1.01i
Virtex-5™	XC5VLX30-3	251	599	81	1	1	0	0	N/A	N/A	ISE 9.1.01i

Notes:

- 1) 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.
- 2) Assuming all core I/Os and clocks are routed off-chip

Applications

Modular Exponentiation is a key mathematical function required by many commonly used Public-Key encryption algorithms such as RSA, Diffie-Hellman, and the Digital Signature Algorithm (DSA) described in FIPS 186-2. These algorithms provide the strong encryption required to provide key exchange and certificate-based authentication for communication protocols such as TLS/SSL and IPsec which are widely used for securing transactions over insecure open networks such as the Internet.

General Description

The Helion ModExp core performs the Modular Exponentiation computation $Z = Y^E \bmod M$ which is at the heart of many commonly used Public-Key encryption schemes such as RSA, Diffie-Hellman and DSA. Modular Exponentiation is an extremely CPU intensive computation which can present a significant overhead for embedded systems which implement these Public-Key algorithms. The Helion ModExp core provides an ideal resource efficient means to perform hardware acceleration for applications which require a cryptographic key exchange.

Functional Description

Figure 1. shows the block diagram of the Helion ModExp Core. It consists of an Operand RAM which provides the data interface and holds the computation operands (Y,E,M) and result (Z), a Modular Multiplier which provides the main datapath and performs the computation, and a Controller block which provides the control interface as well as overseeing the operation of the Multiplier.

Operation of the core is extremely simple. The Y, E and M operands are first written to the Operand RAM by the User application. The operand length is selected and the computation started. Progress of the computation is indicated by the busy and done status outputs from the engine. Once the computation is complete the User application may read the resulting Z value from the RAM.

Core Modifications

The Helion ModExp core may optionally be supplied as a hardwired version supporting Diffie-Hellman Oakley Groups 1, 2, 14 or 15 for use with the Internet Key Exchange (IKE). This removes the need for the user to set up the Modulus value, and in some variants of the core it can reduce the required logic resources. Please contact Helion for further details.

Data Throughput Capability

The Helion Modular Exponentiation core is a scaleable design, and so is available in a choice of versions, each sharing an identical interface, but differing in terms of the numbers of clock cycles they take to perform each operation. This allows the user to size an appropriate solution for any given requirement, trading off performance and logic area.

Tables 1 to 4 show the logic utilisation and maximum clock rates for four of our most popular variants. The smallest core (in Table 1) is called TINY32, and typically offers between 1 and 4 1024-bit RSA operations per second (E=1024, M=1024) depending on your choice of Xilinx target technology. It is therefore a good choice for setting up a small number of secure links in a typical terminal unit application.

For higher performance requirements, the STD64 version covers the range 4 to 10 operations per second, the STD128 version covers the range 10 to 20 operations per second, and the STD256 version covers the range 15 to 40 operations per second; again the exact figure depending on the maximum attainable clock speed in your choice of Xilinx technology.

Modular Exponentiator (ModExp)

Note that all these quoted operation rates are for full-size 1024-bit RSA (E=1024, M=1024). Operations with shorter exponents like those typically used for Diffie-Hellman or for public key encryptions will be much faster in any given implementation, and if evaluating different solutions it is important to ensure that comparisons are made under identical conditions. For accurate performance figures for any of these solutions in any target technology, and for any specific application, please contact Helion and we will be very happy to discuss the options in 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 5.

Table 5: Core I/O Signals.

Signal	Width	Signal Direction	Description
clk	1	Input	Master clock
reset	1	Input	Master asynchronous reset; 1 = reset
Operand Interface (32-bit)			
operand_select	2	Input	Operand select
operand_indata	32	Input	Operand input data word
operand_index	6	Input	Operand data word index
operand_indata_write	1	Input	Operand input data word write enable
operand_outdata	32	Output	Operand output data word
operand_outdata_read	1	Output	Operand output data word read enable
Core Control & Status			
start_processing	1	Input	Start ModExp core
operand_length	2	Input	Operand length (select between 768, 1024, 1536 or 2048-bit operands)
engine_busy	1	Output	ModExp core is busy
engine_done	1	Output	ModExp core is finished (pulse)

Verification Methods

The Helion ModExp core has been thoroughly verified under simulation, using a suite of Helion generated test vectors. In addition, it has been fully proven in real Xilinx hardware, and has already been deployed by our customers within real products out in the field.

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 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

Export

Strong encryption technology, such as the use of Modular Exponentiation in Public-Key Encryption algorithms such as RSA and Diffie-Hellman, is governed internationally by export regulations. Immediate export is permitted to the following countries;

Austria	Australia	Belgium
Bulgaria	Canada	Cyprus
Czech Republic	Denmark	Estonia
Finland	France	Germany
Greece	Hungary	Ireland
Italy	Japan	Latvia
Lithuania	Luxembourg	Malta
New Zealand	The Netherlands	Norway
Poland	Portugal	Romania
Slovakia	Slovenia	Spain
Sweden	Switzerland	United Kingdom
United States		

Please contact Helion to discuss delivery to other destinations; approval is subject to the applicable export licenses being granted. Please note that licensees are responsible for complying with the applicable requirements for re-export of electronics containing strong encryption technology.

Related Information

For more information on Modular Exponentiation and its use in Public-Key cryptography please refer to the following :

- Wikipedia - Modular Exponentiation; http://en.wikipedia.org/wiki/Modular_exponentiation
- Wikipedia - Public Key Cryptography; http://en.wikipedia.org/wiki/Public-key_cryptography
- R.Rivest, A.Shamir, L.Adleman "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems"; <http://theory.lcs.mit.edu/~rivest/rsapaper.pdf>
- RFC2631 "Diffie-Hellman Key Agreement Method"
- RFC2409 "The Internet Key Exchange (IKE)"
- RFC3526 "More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange"
- Digital Signature Standard FIPS 186-3; http://csrc.nist.gov/publications/drafts/fips_186-3/Draft-FIPS-186-3%20_March2006.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