

RC4 is the cipher that supports Secure Sockets Layer (SSL) and the companion Transport Layer Security (TLS) standard. As such, RC4 has been in use for many years and has been proven as a hardened, highly secure cipher. Elliptic has proven the design in both the ELP-10 and ELP-11 virtual components and as such, this semiconductor IP core can be used with confidence by designers.

Key Features:

- Throughput up to 800 Mbps
- Area: 6,000 ASIC gates for encrypt/decrypt, key expansion and interface logic
- Input memory block with shadow memory to allow input of next block while another RC4 operation is in progress
- Key context memory with shadow context for fast key switching
- Automatic generation of key context from key data
- Key memory accessible through separate memory interface
- Support for 40 and 128 bit keys
- Up to 3 shadow memory blocks for input, output and key contexts
- Flow-through or co-processor data flow

Applications:

- SSL/TLS applications
- WLAN WEP, WPA and 802.11i
- Storage – SAN/NAS applications
- Military communications systems
- Secure video surveillance
- Secure audio communications

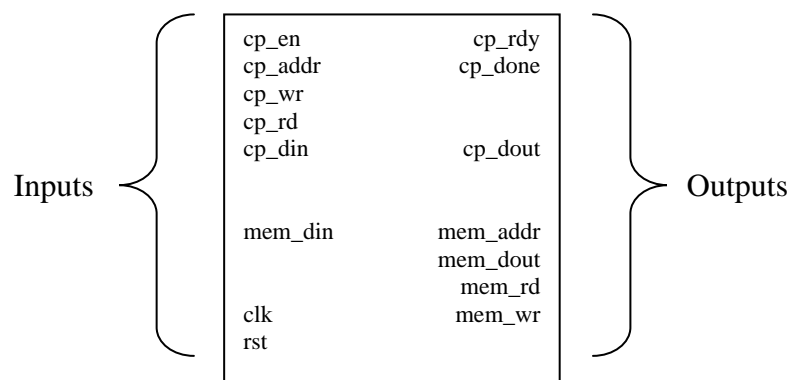


Figure 1 RC4 pin diagram

Pin Description

Signal Name	Width (Bits)	Direction	Description
<i>System Pins</i>			
clk	1	input	System clock. All inputs are sampled on the rising edge of the clock.
rst	1	input	Asynchronous reset. All internal state returns to default value upon assertion (active high) of this signal.
<i>Control Processor Interface</i>			
cp_en	1	input	Enable pin for the interface. All other inputs are ignored unless this pin is asserted.
cp_addr	13	input	Address to perform a read or write. This bus must be valid any time cp_wr or cp_rd (qualified with cp_en) are asserted. The address must always be aligned to a 32-bit access. (Lower 2 bits must be zero).
cp_wr	1	input	Write signal for the data on cp_din and address on cp_addr.
cp_rd	1	input	Read signal for the address on cp_addr.
cp_din	32	input	Data to write. This bus must be valid any time cp_wr (qualified with cp_en) is asserted.
cp_dout	32	output	Result of a read. The value is qualified with cp_rdy.
cp_rdy	1	output	The completion of a read or write transaction is indicated by the cp_rdy line. All cp_* inputs must be held steady until the cycle cp_rdy asserts.
cp_done	1	output	This signal indicates that the encryption has completed. It can be used as an interrupt source to the control processor. It remains asserted until another encryption operation is started.
<i>Memory Interface</i>			
mem_din	32	input	Result of a read operation. Must be valid one cycle after assertion of mem_rd.
mem_addr	-	output	Address of a read or write cycle. Valid whenever mem_wr or mem_rd is asserted. The width of this bus is configurable at build time. (See the RC4_SRC_ADDR and RC4_DST_ADDR registers) All addresses output by the RC4 engine are 32-bit word aligned.

Signal Name	Width (Bits)	Direction	Description
mem_be	4	output	Indicates which bytes are valid on the mem_dout signal when a mem_wr is asserted. The use of these signals is optional. The RC4 engine may be interfaced to a memory without a byte enable port provided the user is not concerned about potential byte over-write of the first and last words of the target memory, which may occur if the data is not 32-bit word aligned in the target memory.
mem_dout	32	output	Write value. Valid when mem_wr is asserted.
mem_rd	1	output	Read strobe. The result is expected on mem_din the next cycle.
mem_wr	1	output	Write strobe. The value of mem_dout is written to the address indicated by mem_addr.

Table 1: Pin description table.

General Description

The RC4 algorithm is a variable key size stream cipher. There are two parts to the algorithm: key expansion, and encryption. The encryption is performed by XORing a stream of pseudo random bytes with the plaintext bytes. The algorithm is perfectly symmetric. By XORing the same pseudo random bytes as used during encryption, the plaintext is recovered from the ciphertext.

The CLP-04 implementation supports the two most common key sizes; 40 and 128 bits. Both the key expansion and encryption parts of the algorithm are implemented. The block also supports a pseudo-random generation mode which just generates the XOR key stream.

The RC4 context consists of a 256 byte expanded key and two single byte i and j pointers. The context is accessed through the control processor port on the block. The context may be automatically generated given a 40 bit or 128 bit key. There are several context pages (configurable number at build time) to allow efficient multiplexing of unrelated data streams to be encrypted.

There are two major interfaces which must be connected. The control processor interface is a slave memory bus which allows an external processor to access the internal configuration registers. The memory interface is a master memory bus which accesses the data to be processed. The memory interface requires single cycle access to a dedicated memory.

The CLP-04 supports a number of configuration options and operations. These are:

- Flush request – clears internal states, plain-text memory, cipher memory and key contexts
- Key-Only mode – ignores input data and generates key stream directly
- Encrypt or decrypt mode
- Swap request between main and shadow memory blocks
- Flow-through or co-processor mode
- Start/finish signals triggers encryption and signals completion.

The CLP-04 is available in soft IP form, either as a Netlist or HDL Source. The deliverables available are:

Netlist Licenses:

- Post synthesis netlist
- Testbench
- Sample simulation script
- Documentation

HDL Source Licenses:

- Verilog HDL
- Testbench
- Sample synthesis script
- Documentation

An FPGA load is available under license for evaluation purposes. For more information on pricing and a full data sheet, please contact:

Elliptic Semiconductor Inc.
62 Steacie Drive, Suite 201
Ottawa, ON, K2K 2A9

Phone: +1 613 254-5456
Fax: +1 613 254-7260
Email: info@ellipticsemi.com