

Functional Description

DES3 is simply a concatenation of three Data Encryption Standard (DES) algorithm operations.

DES Description

The Data Encryption Standard (DES) is a cryptographic algorithm specified in the United States Federal Information Processing Standards Publications (FIPS) 46-3 publication. This publication provides a complete description of a mathematical algorithm for encrypting and decrypting binary coded information.

Encrypting converts the data to an unintelligible form called ciphertext. Decrypting the ciphertext converts the data back to its original form and is called plaintext. The algorithm described in the FIPS 46-3 publication specifies both encrypting and decrypting operations based on a binary number called a key.

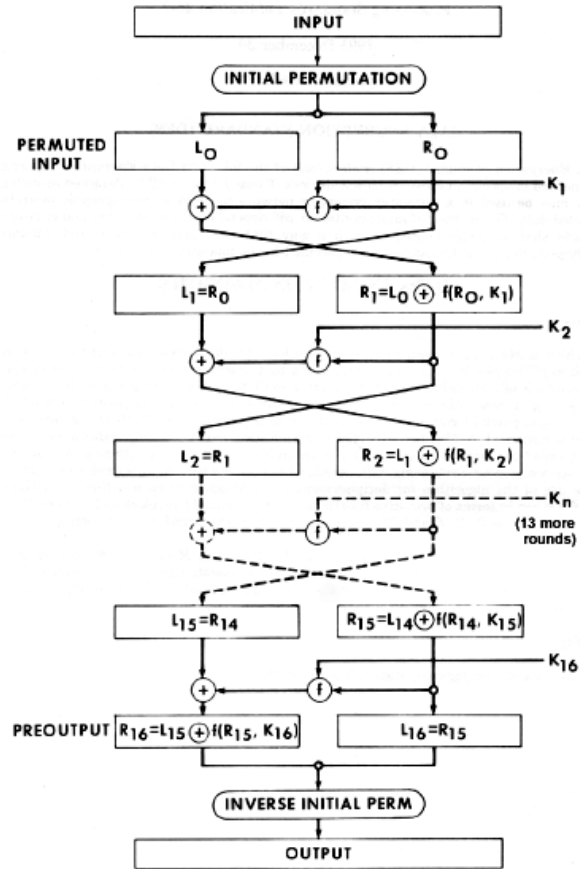
A key consists of 64 binary bits of which 56 bits are usually randomly generated and used directly by the algorithm. The other 8 bits are not used.

Authorized users of the data must have the key that was used to encrypt the data in order to decrypt it. The unique key chosen for use in a particular application makes the results of encrypting data using the algorithm unique. Selection of a different key causes the ciphertext that is produced for any given set of inputs to be different. Therefore, the cryptographic security of the data depends on the security provided for the key used to encrypt and decrypt the data.

DES Block Cipher Encryption Operation

During each pass of the DES engine (module), a 64 bit block is subjected to an initial permutation, then to a complex key-dependent computation and finally to a permutation that is the inverse permutation.

The key-dependent computation is implemented as a function called the key schedule and uses the subkeys generated in the key_sel module. The DES engine can be run in two directions - as a forward transformation and as an inverse transformation. The two directions differ only by the order in which the bits of the key are used. The forward transformation is shown in the following figure.



DES Electronic Code Book (ECB) Mode

The Electronic Codebook (ECB) mode is a block, encryption/decryption method which transforms 64 bits of input to 64 bits of output. The ECB output block is a complex function of all 64 bits of the input block and all 56 independent bits of the key.

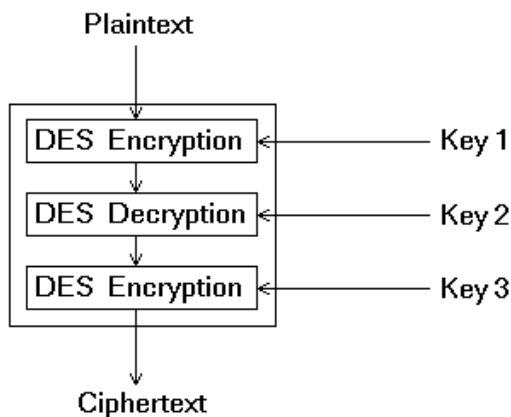
Since the ECB mode is a 64-bit block cipher, the DES engine encrypts data in multiples of sixty-four bits. Therefore a block input is 64 bits wide (Din [63:0]). See signal definitions.

If a user has less than sixty-four bits to encrypt, then the least significant bits of the unused portion of the input data block must be padded. These bits are often filled with random or pseudo-random bits, prior to ECB encryption. The corresponding DES decryption discards these padding bits after decryption of the ciphertext block.

The same input block always produces the same output block under a fixed key in ECB mode.

DES3 Block Cipher Encryption Operation

The procedure for encryption is exactly the same as regular DES, except it is passed through the DES engine three times. The first pass is a DES encryption, the second pass is a DES decryption of the first DES Ciphertext result and the third pass is a DES encryption of the second pass result. This produces the resultant DES3 Ciphertext.



Consequently, DES3 runs three times slower than standard DES, but is much more secure.

The procedure for decrypting a DES3 ciphertext is the same as DES3 encryption except in reverse order. In other words, the process is a DES decryption, DES encryption, DES decryption process using the same keys in reverse order as used in encrypting the ciphertext.

Note that although the input key for DES is 64 bits long, the actual key used by DES is only 56 bits in length. This means that the effective key strength for DES3 is actually 168 bits.

DES3 Mode of Operation = Triple ECB (Electronic Code Book)

The DES3 core works exactly the same way as the Electronic Code Book mode of the DES core. Each pass through the DES module produces an ECB ciphertext to either be used as the input on the next pass or as the resultant DES3 ciphertext.

Implementation Results

DES3 core reference designs have been evaluated in a variety of technologies. The following are sample ASIC results.

Non Pipelined Optimized for Speed					
ASIC Technology	Cell Area	NAND2 Area Used	Approx. Area (gate equiv.)	Frequency (MHz)	Throughput (Gbps)
TSMC .09 μm	15,383	2.82	5,455	909	3.63
TSMC .13 μm	36,215	5.0922	7,112	581	2.32
TSMC .18 μm	71,101	9.9792	7,124	469	1.87
Pipelined Optimized for Speed					
ASIC Technology	Cell Area	NAND2 Area Used	Approx. Area (gate equiv.)	Frequency (MHz)	Throughput (Gbps)
TSMC .09 μm	282,276	2.82	100,098	1,111	71.10
TSMC .13 μm	747,667	5.0922	146,825	787	50.36
TSMC .18 μm	1,367,393	9.9792	137,024	591	37.82

Example of DES3 Encryption AND Decryption Operations

The following is an example that may be used when testing the DES3 encryption and decryption operations. In this example, all keys, plaintext and ciphertext are expressed in hexadecimal. The example uses three independent keys, which are:

Key1 = 0123456789ABCDEF

Key2 = 23456789ABCDEF01

Key3 = 456789ABCDEF0123

The plaintext for the example is selected from the ASCII encoding of the phrase “The quick brown fox jumped over the lazy dog’s back”. The example employs the first 24 characters of the phrase (i.e., The quick brown fox jump).

The ASCII encoding of the above phrase is segmented as follows:

“The quic”	5468652071756663
“k brown ”	6B2062726F776E20
“fox jump”	666F78206A756D70

DES3 Block Cipher Encryption Operation - ECB Mode

In the example below, the input and output of the DES engine are given sequentially.

Note that DES1, DES2 and DES3 represent the three passes through the DES engine.

The input to DES1 is PlainText P1, and the output of DES1 is “A28E91724C4BBA31”. The input to DES2 is the output of DES1, and the output of DES2 is “5A2EA7F983A2F53F”. The input to DES3 is the output of DES2, and the output of DES3 is “A826FD8CE53B855F”. The output of DES3 is the ciphertext C1.

P1 = “The quic” = 5468652071756663

	Input	Output
DES1 – Encrypt – Key1	5468652071756663	A28E91724C4BBA31
DES2 – Decrypt – Key2	A28E91724C4BBA31	5A2EA7F983A2F53F
DES3 – Encrypt – Key3	5A2EA7F983A2F53F	A826FD8CE53B855F

C1 = A826FD8CE53B855F

During the second DES3 operation, the input is P2, and the output after the three passes is ciphertext C2.

P2 = “k brown ” = 6B2062726F776E20

	Input	Output
DES1 – Encrypt – Key1	6B2062726F776E20	167E47EC24F71D63
DES2 – Decrypt – Key2	167E47EC24F71D63	EA141A7DD69701F0
DES3 – Encrypt – Key3	EA141A7DD69701F0	CCE21C8112256FE6

C2 = CCE21C8112256FE6

During the third DES3 operation, the input is P3, and the output after the three passes is ciphertext C3.

P3 = " fox jump" = 666F78206A756D70

	Input	Output
DES1 – Encrypt – Key1	666F78206A756D70	2C1A917234425365
DES2 – Decrypt – Key2	2C1A917234425365	8059EE8212E22A79
DES3 – Encrypt – Key3	8059EE8212E22A79	68D5C05DD9B6B900

C3 = 68D5C05DD9B6B900

The resulting ciphertext is the concatenation of C1, C2 and C3 (i.e., A826FD8CE53B855F CCE21C8112256FE6 68D5C05DD9B6B900).

DES3 Block Cipher Decryption Operation - ECB Mode

During decryption operations in ECB mode, the ciphertext C1, C2 and C3 (section from 3.1) are fed into the DES3 to produce the plaintext P1, P2 and P3. The output of DES1 becomes the input to DES2, and the output of DES2 becomes the input to DES3.

C1 = A826FD8CE53B855F

	Input	Output
DES1 – decrypt - Key3	A826FD8CE53B855F	5A2EA7F983A2F53F
DES2 – encrypt - Key2	5A2EA7F983A2F53F	A28E91724C4BBA31
DES3 – decrypt - Key1	A28E91724C4BBA31	5468652071756663

P1 = 5468652071756663 = "The quic".

C2 = CCE21C8112256FE6

	Input	Output
DES1 – decrypt – Key3	CCE21C8112256FE6	EA141A7DD69701F0
DES2 – encrypt – Key2	EA141A7DD69701F0	167E47EC24F71D63
DES3 – decrypt – Key1	167E47EC24F71D63	6B2062726F776E20

P2 = 6B2062726F776E20 = "k brown "

C3 = 68D5C05DD9B6B900

	Input	Output
DES1 – decrypt – Key3	68D5C05DD9B6B900	8059EE8212E22A79
DES2 – encrypt – Key2	8059EE8212E22A79	2C1A917234425365
DES3 – decrypt – Key1	2C1A917234425365	666F78206A756D70

P3 = 666F78206A756D70= "fox jump".

The plaintext is the ASCII encoding of "The quick brown fox jump".

Support

The core as delivered is warranted against defects for ninety days from purchase. Thirty days of phone and email technical support are included, starting with the first interaction. Additional maintenance and support options are available.

Verification

The core has been verified through extensive simulation and rigorous code coverage measurements.

Deliverables

The core includes everything required for successful implementation. The ASIC version includes:

- HDL RTL source code
- Sophisticated self-checking Testbench
- Simulation script, vectors, and expected results
- Synthesis script
- Comprehensive user documentation