DES Cores

Introduction

The HammerCores by Altera® library of DES encryption and decryption cores consists of:

- DES encryption core
- DES decryption core
- DES encryption / decryption core (control bit selectable)

The cores are compact – 450 to 600 Logic Cells (LCs), and high performance – up to 125 Mbps The cores may be used with Altera FLEX® 6000, FLEX 8K, and FLEX 10K architectures.

DES Standard

These cores are compliant with DES (Data Encryption Standard), as described in ***Federal Information Processing Standards (FIPS) Publication 46-2, of January 15,1977***. Note that these cores have not been validated by any third party.

DES encrypts 64-bit blocks, with a 56-bit key, resulting in a 64-bit ciphertext. The key is often given in a 64-bit word, of which 8 are parity bits, at locations 8, 16, 24 … 64. Decryption of the ciphertext will only be successful with the identical key used for encryption.

There are several DES operating modes defined, such as ECB, CBC, and CFB. These may be supported by adding additional logic to the core inside a programmable logic device. DES modes of operation are explained later in this document.

Export Restrictions

DES devices and implementations, as well as systems incorporating DES, are subject to export controls. These cores will not be distributed or supported outside of the United States or Canada.

DES Encryption Core

The DESENC core is a DES encryption core for Altera programmable logic devices. The core encrypts a 64-bit block, using a 64-bit key, and returns a 64-bit block of ciphertext. All interfaces, for input plaintext, output ciphertext, and the key, are 64-bits wide. The user can add external logic to control the input and output of data to the core, in a serial, byte, or word wide format.

The DES encryption core is created by compiling the **DESENC.TDF** file.

Interfaces

Table 1. Input Signals

Signal Name	Description
SYSCLK	SYSCLK is the main system clock. After resetting the DES core, 17 clock cycles are required to encrypt a 64-bit plain text block.
RESET	RESET is an asynchronous, active high signal, which resets all internal registers of the core, and prepares the core to encrypt another plaintext block.
INWORD[64..1]	This is the 64-bit plaintext input block. All 64 bits must be present after RESET is deasserted.
KEY[64..1]	The 64-bit key input word consists of a 56-bit key, with 8 parity bits. The paritybits are at locations 8, 16, 24, 32, 40, 48, 56, and 64. The DES core performs no parity checking of the bits. The key may be changed at any time.

Table 2.Output Signals

Signal Name	Description
DONE	This signal is asserted (active high) when the encryption operation is complete.
OUTWORD[64..1]	This bus contains the ciphertext when DONE is asserted.

Operation

All 64 bits of the plaintext input, and all 64 bits of the key must be present before encryption begins. Encryption begins at the next rising edge of SYSCLK after the RESET signal is deasserted. After 17 clock cycles, DONE is asserted (active high), signifying that a ciphertext for the plaintext input has been presented on the OUTWORD[] bus.

After the *second* clock cycle of the encryption operation, the plaintext and key inputs can be changed without affecting the encryption operation.

Endian

When you use the DES core, especially when interfacing with other DES implementations, the correct bit ordering must be used. DES specifies that the 64-bit input and output words are arranged from bit location 1 (left most, most significant) to bit location 64 (right most, least significant).

As an example, the key 0123456789ABCDEF will be shown in the bit order, from left to right.

0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

When using the Altera simulator, which treats the vector as [64..1], rather than [1..64], this would be input as:

1111 0111 1011 0011 1101 0101 1001 0001 1110 0110 1010 0010 1100 0100 1000 0000, or

F7B3D591E6A2C480, in HEX mode.

The bit ordering for each nibble is corrected using the **ENDIAN.TDF** file. The **XDESENC.TDF** design file applies the endian correction to the DESENC design. As the Altera simulator still orders the bits as [64..1], the key input is now FEDCBA987654310.

Verification

The DESENC core was tested with several known sequences, as well as in a back-to-back encryption / decryption mode with the decryption core, DESDEC.

The following test sequences are taken from ***Federal Information Processing Standards (FIPS) 1981***, page 18. Note that they have been re-ordered for the [64..1] input, but use the **XDESENC.TDF** design, so that each nibble has the same hexadecimal representation as in the ***Federal Information Processing Standards (FIPS) Publication 46-2, of January 15,1977***.

Key: FEDCBA9876543210

Table 3. Input and Output Values

DES Input (PLAINTEXT)	DES Output (CIPHERTEXT)
FEDCBA0987654321	52E478EA965166DB
FDB975121FCA8642	01A7CAF1C9613B84
FB73FA242E951D84	EF9599C4933410A0
E7F6E5584C3B2A19	11720B8DF55F25D6
CFEDCBA098765432	298AF2A3BBED83A3
9FDB975121FCA864	CCA7ECD3DB07B917
3FB73FA242E951D8	32D0CDA032C90818
6E7F6E5584C3B2A1	4066296AD6A41D38
DCFEDCBA09876543	A8D25D6D8CD9E113
A9FDB975121FCA86	2724CF6BEF7C74BD

Implementation

The DESENC core was compiled into Altera FLEX 6000 and FLEX 10K devices. The logic synthesis option should be set to FAST. Table 4 summarizes the results.

Table 4. Implementation Results

Device	Size	Performance (MHz)	Performance (Mbps)
EPF10K30A-1	467 LCs	53 MHz	188 Mbps
EPF10K30A-3	467 LCs	36 MHz	128 Mbps

Note:
Performance in Mbps is based on 18 clock cycles per encryption operation, including reset of the DES core.

DES Decryption Core

DESDEC

The DESDEC core is a DES decryption core for Altera programmable logic devices. The core decrypts a 64-bit block of ciphertext, using a 64-bit key, and returns a 64-bit block of plaintext. All interfaces, for input ciphertext, output plaintext, and the key, are 64-bits wide. The user can add external logic to control the input and output of data to the core, in a serial, byte, or word wide format.

The DES decryption core is created by compiling the **DESDEC.TDF** file.

Interfaces

Table 5. Input Signals

Signal Name	Description
SYSCLK	SYSCLK is the main system clock. After resetting the DES core, 17 clock-cycles are required to decrypt the 64-bit ciphertext block.
RESET	RESET is an asynchronous, active high signal, which resets all internal registers of the core, and prepares the core to decrypt another ciphertext block.
INWORD[64..1]	This is the 64-bit ciphertext input block. All 64 bits must be present after RESET is deasserted.
KEY[64..1]	The 64 bit key input word consists of a 56-bit key, with 8 parity bits. The paritybits are at locations 8, 16, 24, 32, 40, 48, 56, and 64. The DES core performs no parity checking of the bits. The key may be changed at any time.

Table 6. Output Signals

Signal Name	Description
DONE	This signal is asserted (active high) when the decryption operation is complete.
OUTWORD[64..1]	This bus contains the plaintext when DONE is asserted.

Operation

All 64 bits of the ciphertext input, and all 64 bits of the key must be present before decryption begins. Decryption begins at the next rising edge of SYSCLK after the RESET signal is deasserted. After 17 clock cycles, DONE is asserted (active high), signifying that plaintext for the ciphertext input has been presented on the OUTWORD[] bus.

After the second clock cycle of the decryption operation, the ciphertext and key inputs can be changed without affecting the decryption operation.

Endian

A very important consideration in using the DES core, especially when interfacing with other DES implementations, is using the correct bit ordering. DES specifies that the 64-bit input and output words are arranged from bit location 1 (left most, most significant) to bit location 64 (right most, least significant).

As an example, the key 0123456789ABCDEF will be shown in the bit order, from left to right.

0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

When using the Altera simulator, which treats the vector as [64..1], rather than [1..64], this would be input as:

1111 0111 1011 0011 1101 0101 1001 0001 1110 0110 1010 0010 1100 0100 1000 0000, or

F7B3D591E6A2C480, in HEX mode.

The bit ordering for each nibble is corrected using the **ENDIAN.TDF** file. The **XDESDEC.TDF** design file applies the endian correction to the DESDEC design. As the Altera simulator still orders the bits as [64..1], the key input is now FEDCBA987654310.

Verification

The DESDEC core was tested will several known sequences, as well as in a back-to-back encryption / decryption mode with the encryption core, DESENC.

The following test sequences are taken from in ***Federal Information Processing Standards (FIPS) Publication 46-2, of January 15,1977***, page 18. Note that they have been re-ordered for the [64..1] input, but use the **XDESDEC.TDF** design, so that each nibble has the same hexadecimal representation as in the in ***Federal Information Processing Standards (FIPS) Publication 46-2, of January 15,1977***. The ciphertext results were also verified by encrypting the plaintext using the **XDESENC.TDF** design.

Key: FEDCBA9876543210

Table 7. Input and Output Values

DES Input (CIPHERTEXT)	DES Output (PLAINTEXT)
52E478EA965166DB	FEDCBA0987654321
01A7CAF1C9613B84	FDB975121FCA8642
EF9599C4933410A0	FB73FA242E951D84
11720B8DF55F25D6	E7F6E5584C3B2A19
298AF2A3BBED83A3	CFEDCBA098765432
CCA7ECD3DB07B917	9FDB975121FCA864
32D0CDA032C90818	3FB73FA242E951D8
4066296AD6A41D38	6E7F6E5584C3B2A1
A8D25D6D8CD9E113	DCFEDCBA09876543
2724CF6BEF7C74BD	A9FDB975121FCA86

Implementation

The DESDEC core was compiled into Altera FLEX 6000 and FLEX 10K devices. The logic synthesis option should be set to FAST. Table 8 summarizes the results.

Table 8. Performance

Device	Size	Performance (MHz)	Performance (Mbps)
EPF10K30A-1	467 LCs	58 MHz	206 Mbps
EPF10K30A-3	467 LCs	38 MHz	135 Mbps

Note:
Performance in Mbps is based on 18 clock cycles per decryption operation, including reset of the DES core.

DES Encryption/Decryption Core

The DES Encryption / Decryption Core (ENCDEC) core is a DES encryption / decryption core for Altera programmable logic devices. The mode, either encryption or decryption, is selected by an external control pin. The core operates on a 64-bit block, using a 64-bit key, and returns a 64-bit block of ciphertext, or plaintext, depending on the mode selected. All interfaces, for input, output, and the key, are 64-bits wide. The user can add external logic to control the input and output of data to the core, in a serial, byte, or word wide format.

The DES encryption / decryption core is created by compiling the **ENCDEC.TDF** file.

Interfaces

Table 9. Input Signals

Signal Name	Description
SYSCLK	SYSCLK is the main system clock. After resetting the DES core, 17 clock cycles are required to encrypt or decrypt the 64 bit input block.
RESET	RESET is an asynchronous, active high signal, which resets all internal registers of the core, and prepares the core to encrypt or decrypt another block.
ENCRYPT	When high, the core encrypts the input block. When low, the core decrypts the input block.
INWORD[64..1]	This is the 64 bit input block. All 64 bits must be present after RESET is deasserted.
KEY[64..1]	The 64-bit key input word consists of a 56-bit key, with 8 parity bits. The paritybits are at locations 8, 16, 24, 32, 40, 48, 56, and 64. No parity checking of the bits is performed by the DES core. The key may be changed at any time.

Table 10. Output Signals

Signal Name	Description
DONE	This signal is asserted (active high) when the current operation is complete.
OUTWORD[64..1]	This bus contains the result when DONE is asserted.

Operation

All 64 bits of the input, and all 64 bits of the key must be present before an encryption or decryption operation begins. The operation begins at the next rising edge of SYSCLK after the RESET signal is deasserted. After 17 clock cycles, DONE is asserted (active high), signifying that data on the OUTWORD[] bus is valid.

After the second clock cycle of the decryption operation, the ciphertext and key inputs can be changed without affecting the decryption operation. The ENCRYPT signal must remain constant during the selected operation.

Endian

A very important consideration in using the DES core, especially when interfacing with other DES implementations, is using the correct bit ordering. DES specifies that the 64 bit input and output words are arranged from bit location 1 (left most, most significant) to bit location 64 (right most, least significant).

As an example, the key 0123456789ABCDEF will be shown in the bit order, from left to right.

0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

When using the Altera simulator, which treats the vector as [64..1], rather than [1..64], this would be input as:

1111 0111 1011 0011 1101 0101 1001 0001 1110 0110 1010 0010 1100 0100 1000 0000, or

F7B3D591E6A2C480, in HEX mode.

The bit ordering for each nibble is corrected using the **ENDIAN.TDF** file. The **XENCDEC.TDF** design file applies the endian correction to the ENCDEC design. As the Altera simulator still orders the bits as [64..1], the key input is now FEDCBA987654310.

Verification

The following test sequences are taken from ***Federal Information Processing Standards (FIPS) Publication 46-2, of January 15,1977***, page 18. Note that they have been re-ordered for the [64..1] input, but use the **XENCDEC.TDF** design, so that each nibble has the same hexadecimal representation as in the ***Federal Information Processing Standards (FIPS) Publication 46-2, of January 15,1977***.

Key: FEDCBA9876543210

Table 11. Input & Output Vales

DES Output (PLAINTEXT)	DES Input (CIPHERTEXT)	DES Output (PLAINTEXT)
FEDCBA0987654321	52E478EA965166DB	FEDCBA0987654321
FDB975121FCA8642	01A7CAF1C9613B84	FDB975121FCA8642
FB73FA242E951D84	EF9599C4933410A0	FB73FA242E951D84
E7F6E5584C3B2A19	11720B8DF55F25D6	E7F6E5584C3B2A19
CFEDCBA098765432	298AF2A3BBED83A3	CFEDCBA098765432
9FDB975121FCA864	CCA7ECD3DB07B917	9FDB975121FCA864
3FB73FA242E951D8	32D0CDA032C90818	3FB73FA242E951D8
6E7F6E5584C3B2A1	4066296AD6A41D38	6E7F6E5584C3B2A1
DCFEDCBA09876543	A8D25D6D8CD9E113	DCFEDCBA09876543
A9FDB975121FCA86	2724CF6BEF7C74BD	A9FDB975121FCA86

Implementation

The ENCDEC core was compiled into Altera FLEX 6000 and FLEX 10K devices. The logic synthesis option should be set to FAST. Table 12 summarizes the results:

Table 12. Performance

Device	Size	Performance (MHz)	Performance (Mbps)
EPF10K30A-1	635 LCs	47 MHz	167 Mbps
EPF10K30A-3	635 LCs	33 MHz	117 Mbps

Note:
Performance in Mbps is based on 18 clock cycles per encryption or decryption operation, including reset of the DES core.

DES Operating Modes

One way to attack, or attempt to break, a cipher, is by the known plaintext attack. A brute force method, testing all keys until known plaintext matches the corresponding ciphertext, is sure to work, although costly in time. For DES, this requires checking 2^{56} keys.

Known plaintext may be simple to determine, without knowing the contents of the message, but only the type of data. Most files, such as UNIX files, will have a known header. To counteract the known plaintext attack, several modes of operation are defined, to hide the plaintext.

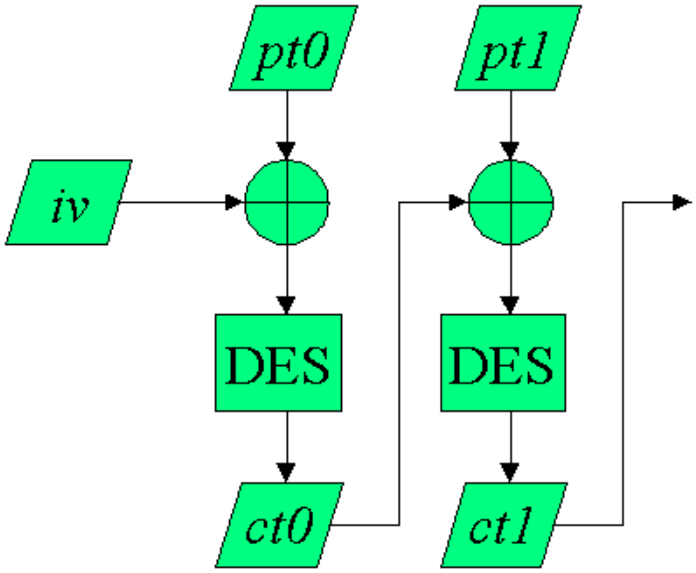
ECB Mode

ECB (Electronic Codebook) mode is nothing more than the DES standard itself. No processing is done before or after encryption.

CBC Mode

CBC (Cipher Block Chaining) mode encryption is illustrated in Figure 1.

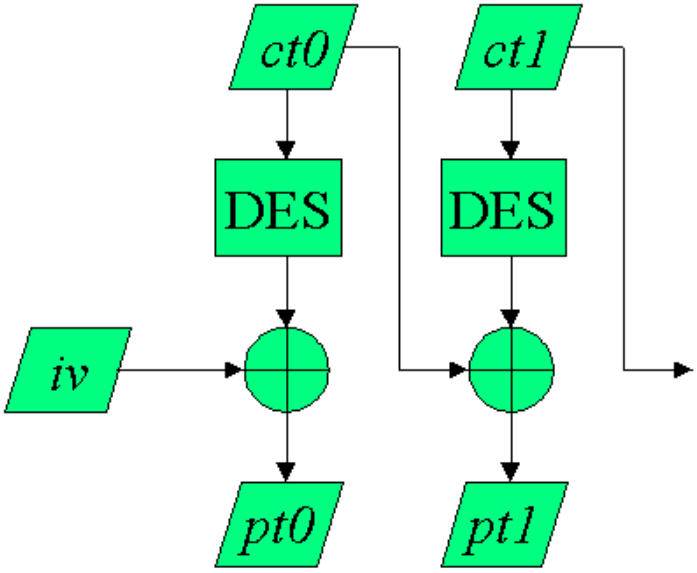
Figure 1. CBC Mode Encryption Flow



The first incoming plaintext is XORed with an initialization vector (*iv*). After encryption, the ciphertext is XORed with the next plaintext vector. Therefore, the result of each encryption is XORed with the next plaintext. This mode can be added easily by the user. A multiplexer can be used to select between the initialization vector and the previous ciphertext for the input to the core; as the core will generally be byte or wordwise loaded, the multiplexer and the XOR function will require a very small amount of logic.

Decryption in the CBC mode is reversed from the encryption, and is shown in Figure 2.

Figure 2. CBC Mode Decryption Flow

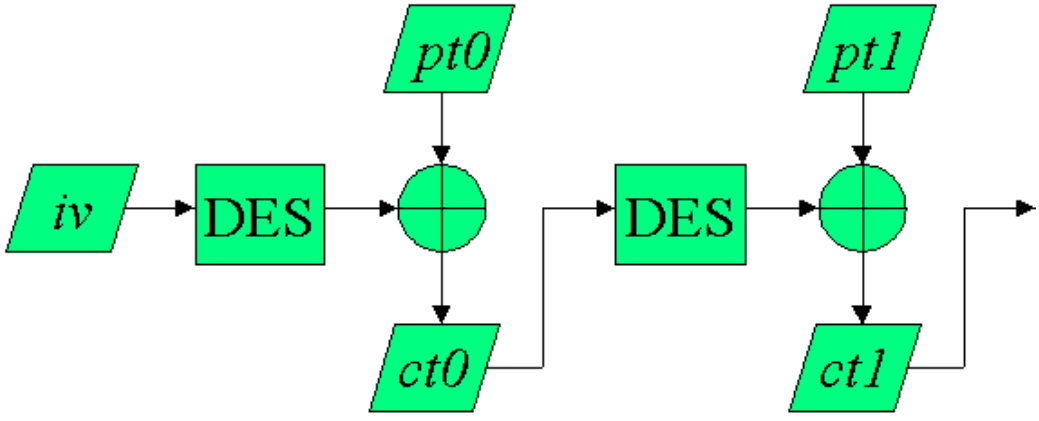


The incoming ciphertext is decrypted before XORing with *iv*, but is used to XOR the decryption result of the next ciphertext. Again, this mode can be implement using only a multiplexer and some XOR logic.

CFB Mode

The CFB mode encryption flow is shown in Figure 3.

Figure 3. CFB Encryption



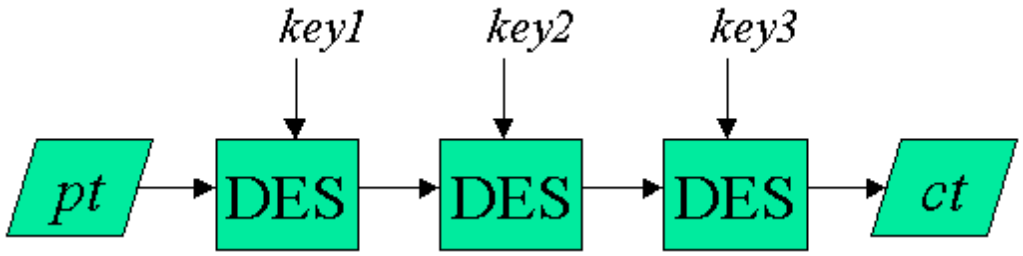
In this mode, *iv*, is encrypted, and XORed with the incoming plaintext, to create the first ciphertext. As with the CBC mode, the implementation is very straightforward, with only a multiplexer and XOR logic needed.

CFB decryption is identical to encryption, except that the plaintext and ciphertext will be swapped. (Incoming ciphertext, outgoing plaintext).

Triple DES

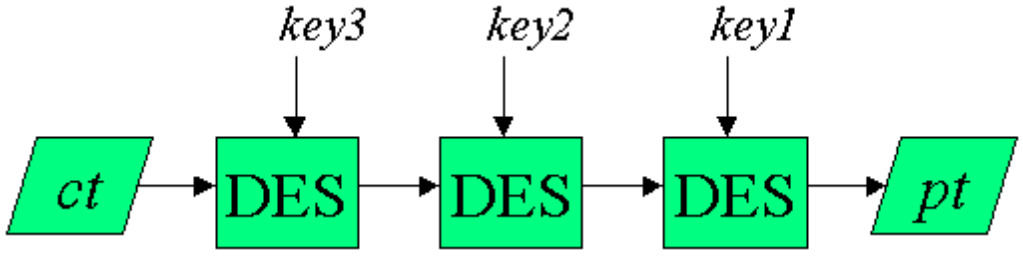
Triple DES is simply three DES Encryptions in a row, preferably with three independent keys. A ECB Triple DES encryption is illustrated in Figure 4. Other modes of operation, such as chaining and feedback modes, may also be used with Triple DES, but are not shown.

Figure 4. Triple DES Encryption



Decryption is the same as encryption, but the order of the keys is reversed.

Figure 5. Triple DES Decryption



101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com

	services. All rights reserved.