CoreDES

Product Summary

Intended Use

- Whenever Data is Transmitted across an Accessible Medium (Wires, Wireless, etc.)
- E-Commerce Transactions, where Dedicated Encryption/Decryption Hardware Can Ease the Load on Servers
- Personal Security Devices
- Bank Transactions, where Financial Security is Mandatory

Key Features

- 56-bit Cipher Key (with 8 Additional Parity Bits)
- Parity Checking Logic for Cipher Key
- Encryption and Decryption Possible with Same Core
- 16-Clock Cycle Operation to Encrypt or Decrypt 64 Bits of Data
- Pause/Resume Functionality to Continue Encryption or Decryption at Will
- Compliant with FIPS PUB 46-3
- ECB (Electronic Codebook) Implementation per FIPS PUB 81
- Example Source Code Provided for CBC, CFB and OFB Modes
- Provides Data Security within a Secure Actel FPGA
- All Major Actel Device Families Supported

Supported Families

- Fusion
- ProASIC3/E
- ProASIC^{PLUS}
- Axcelerator
- RTAX-S
- SX-A
- RTSX-S

Core Deliverables

- Evaluation Version
  - Compiled RTL Simulation Model Fully Supported in the Actel Libero® Integrated Design Environment (IDE)
- Netlist Version
  - Structural Verilog and VHDL Netlists (with and without I/O pads) Compatible with the Actel Designer Place-and-Route Software Tool
  - Compiled RTL Simulation Model Fully Supported in the Actel Libero IDE
- RTL Version
  - Verilog or VHDL Core Source Code
  - Core Synthesis Scripts
- Actel-Developed Testbench (Verilog and VHDL)

Synthesis and Simulation Support

- Synthesis: Synplicity®, Synopsys® (Design Compiler® / FPGA Compiler™ / FPGA Express™), Exemplar™
- Simulation: OVI-Compliant Verilog Simulators and Vital-Compliant VHDL Simulators

Core Verification

- Actel-Developed Simulation Testbench Verifies CoreDES against Tests Listed in the National Institute of Standards and Technology (NIST) Special Publication 800-17, *Modes of Operation Validation System (MOVS): Requirements and Procedures*
- Users Can Easily Modify Testbench Using Existing Format to Add More Tests Listed in NIST Special Publication 800-17 or Custom Tests

Contents

General Description

The CoreDES macro implements the Data Encryption Standard (DES), which provides a means of securing data. The DES algorithm is described in *Federal Information Processing Standards (FIPS) Publication (PUB) 46-3*. The algorithm takes as input 64 bits of plaintext data and 64 bits of a cipher key (only 56 of the 64 bits of the key are used in the calculations, as the least significant bit of each byte of the cipher key is used to provide odd-parity for the key bytes) and after 16 cycles, produces a 64-bit ciphered version of the original plaintext data as output. During the 16 cycles or iterations of the algorithm, the data bits are subjected to permutation and addition functions, which consist of key schedules, calculated by rotations and permutations applied to the original 56-bit cipher key. Figure 1 illustrates the 16-iteration DES algorithm, as described in detail in *FIPS PUB 46-3*.

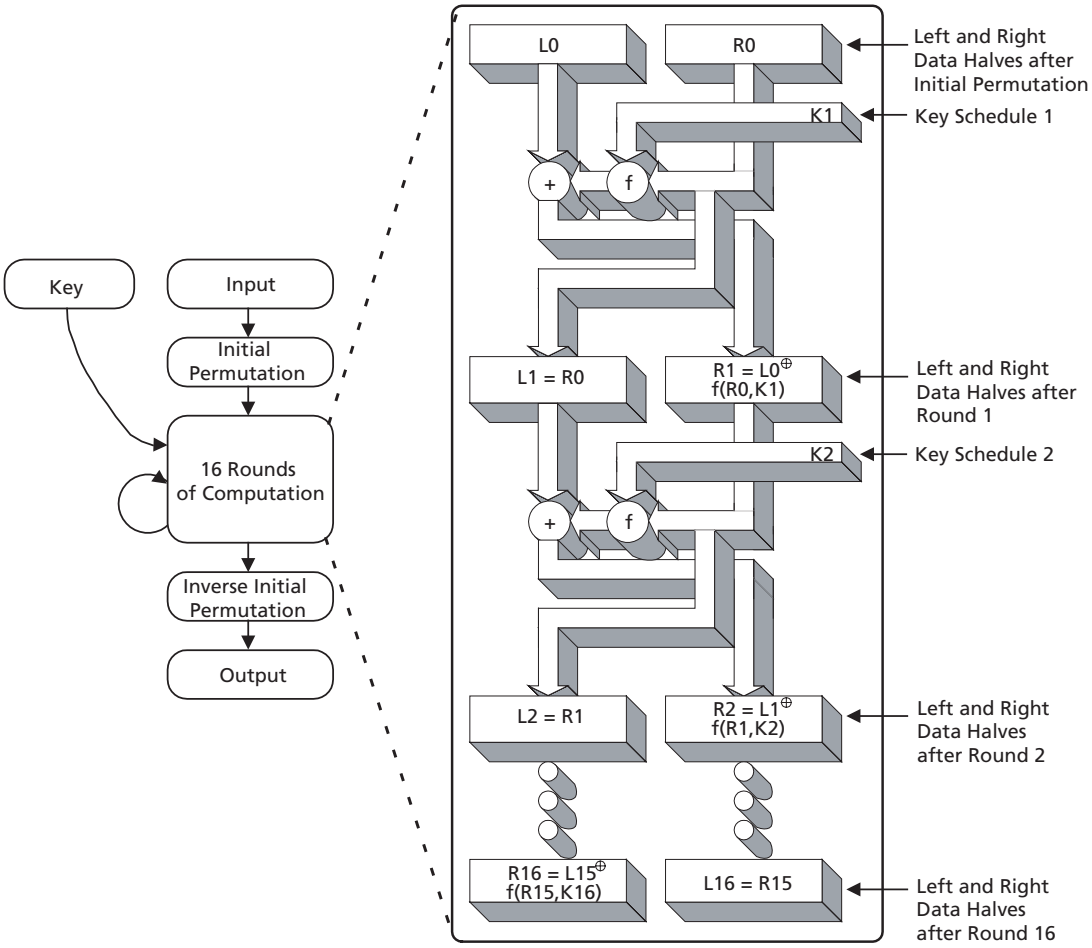

Figure 1 • **DES Algorithm**

CoreDES consists of four main blocks (shown in Figure 2).

1. Data schedule logic – computes the intermediate data values at each round of the DES algorithm.
2. Iteration state machine logic – keeps track of which round of the DES algorithm is currently in progress.
3. Key schedule logic – computes the intermediate keys at each round of the DES algorithm.
4. Parity check logic – checks for odd-parity compliance of the 56 bits of cipher key and issues an error signal if parity is not correct.

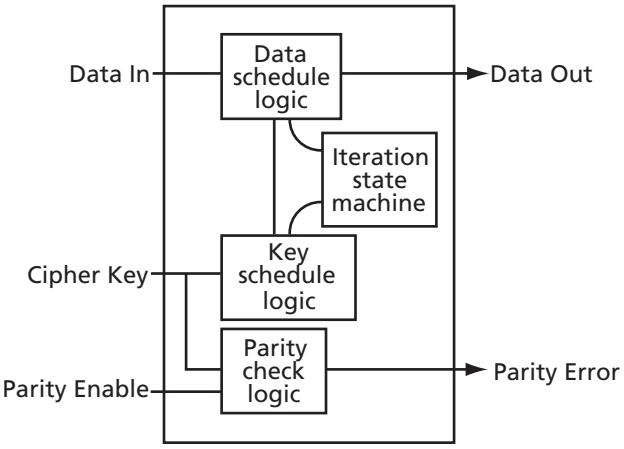

Figure 2 • **CoreDes Block Diagram**

Design Security

Figure 3 shows a typical system diagram. Note that the cipher key, which is the "secret" key, can be made up of FPGA logic cells thereby preventing the possibility of design or data theft. Actel Flash-based devices (ProASIC^{PLUS}) employ FlashLock^{TM} technology, and Actel antifuse-based devices (Axcelerator, SX-A, RT54SX-S) employ FuseLock^{TM} technology, each of which provides a means to keep the cipher key and the rest of the logic secure. The output of the CoreDES macro should be connected to registers or FIFOs, as it is only valid for one clock cycle, as shown in the sections"Encryption" on page 6 and "Decryption" on page 7, respectively.

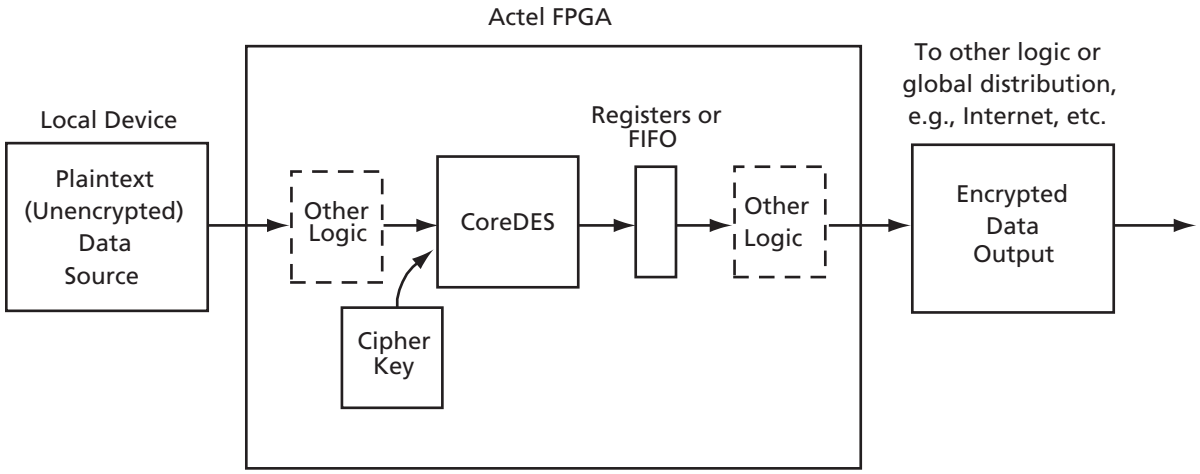

Figure 3 • **CoreDES in Typical System**

CoreDES Device Requirements

The CoreDES macro has been implemented in several of the Actel device families. A summary of the implementation data is listed in Table 1.

Table 1 • **CoreDes Device Utilization and Performance**

Family	Cells or Tiles			Utilization		Performance	Throughput
	Sequential	Combinatorial	Total	Device	Total		
Fusion	148	1123	1271	AFS600	10%	80 MHz	320 Mbps
ProASIC3/E	148	1123	1271	A3PE600-2	10%	80 MHz	320 Mbps
ProASIC^{PLUS}	142	1328	1470	APA075-STD	48%	50 MHz	200 Mbps
Axcelerator	141	601	742	AX125-3	37%	125 MHz	500 Mbps
RTAX-S	141	601	742	RTAX1000S-1	4%	74 MHz	296 Mbps
SX-A	141	628	769	A54SX16A-3	53%	100 MHz	400 Mbps
RTSX-S	141	628	769	RT54SX32S-2	27%	55 MHz	220 Mbps

Note: Data in this table achieved using typical synthesis and layout settings.

Data throughput is computed by taking the bit width of the data (64 bits), dividing by the number of cycles (16), and multiplying by the clock rate (performance); the result is listed in Mbps (millions of bits per second).

CoreDES Verification

The comprehensive simulation testbench (included with Netlist and RTL versions of the core) verifies the CoreDES macro against several of the tests listed in NIST Special Publication 800-17, *Modes of Operation Validation System (MOVS): Requirements and Procedures*. The testbench applies several tests to the CoreDES macro, including: sample round output tests, variable plaintext tests, variable cipher key tests, permutation operation tests, and substitution table tests. Using the supplied testbench as a guide, the user can easily customize the verification of the core by adding or removing any of the tests listed in *NIST Special Publication 800-17* or by adding any custom test cases.

I/O Signal Descriptions

The port signals for the CoreDES macro are defined in Table 2 and illustrated in Figure 4. CoreDES has 200 I/O signals that are described in Table 2. All arrayed ports are labeled with indices that begin with the number 1 (most significant bit) and ascend up to the width of the arrayed port (least significant bit, which happens to be 64 for all arrayed ports in this core). The arrayed ports are labeled in this fashion to correspond with the nomenclature described in *Federal Information Processing Standards Publication 46-3 (FIPS PUB 46-3)*.

Table 2 • **CoreDES I/O Signal Descriptions**

Name	Type	Description
NRESET	Input	Active-low asynchronous reset
CLK	Input	System clock: reference clock for all internal DES logic
EN	Input	Enable signal: set to '1' for normal continuous operation, set to '0' to pause
CLR	Input	Synchronous clear signal: set to '1' to clear logic at any time
ED	Input	Encrypt/Decrypt: '1' to Encrypt, '0' to Decrypt
PCHK	Input	Parity Check: set to '1' to enable parity checking of cipher key bits
K[1:64]	Input	Key: 64-bit (56 bits + 8 parity bits) cipher key input bus
D[1:64]	Input	Data in: 64-bit data input bus
Q[1:64]	Output	Data out: 64-bit ciphertext (for Encrypt operation, plaintext for Decrypt operation)
QVAL	Output	Q Valid: '1' indicates that valid Encrypt/Decrypt data is available on Q
PERR	Output	Parity Error: '1' indicates that a parity error has occurred on the K cipher key input bits

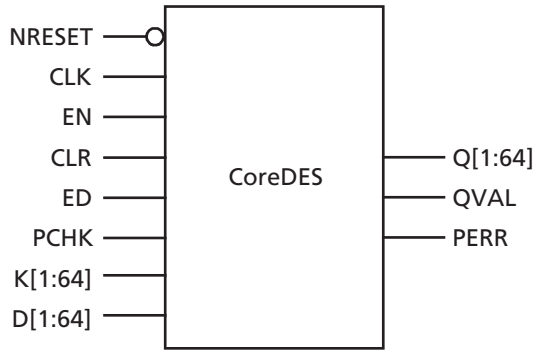

Figure 4 • **CoreDES I/O Signal Diagram**

CoreDES Operation

Parity Checking

If parity checking is desired for the cipher key K[1:64] inputs, the PCHK input should be held at logic '1'. The parity checking logic will determine whether or not an odd number of logic '1' values are present in each byte of the cipher key. This function can be disabled at any time by setting the PCHK input to logic '0'.

Note that if parity checking is disabled by setting the PCHK input to logic '0,' the least significant bits of each byte of the cipher key (K[8], K[16], K[24], K[32], K[40], K[48], K[56], and K[64]) can each be statically connected to either a logic '1' or logic '0' value, since they are the parity bits and will not be used (Figure 5).

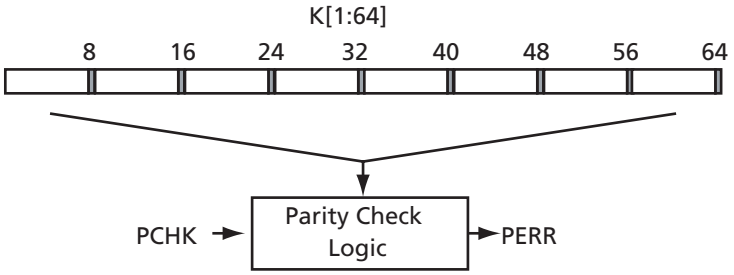

Figure 5 • **Key Parity Check**

Encryption

To begin the process of encrypting data, the following inputs are set:

1. K[1:64] is set to the cipher key (ck1 in Figure 6) to encrypt the data.
2. D[1:64] is set to the plaintext data (d1 in Figure 6) to be encrypted.
3. ED is set to logic '1'.
4. EN is set to logic '1'.

After 16 clock cycles of the EN input being held continuously at a logic '1' value, the QVAL signal will transition from logic '0' to logic '1' and remain valid for one clock cycle, indicating that valid ciphered (encrypted) data (shown as q1 in Figure 6) is available on the Q[1:64] outputs.

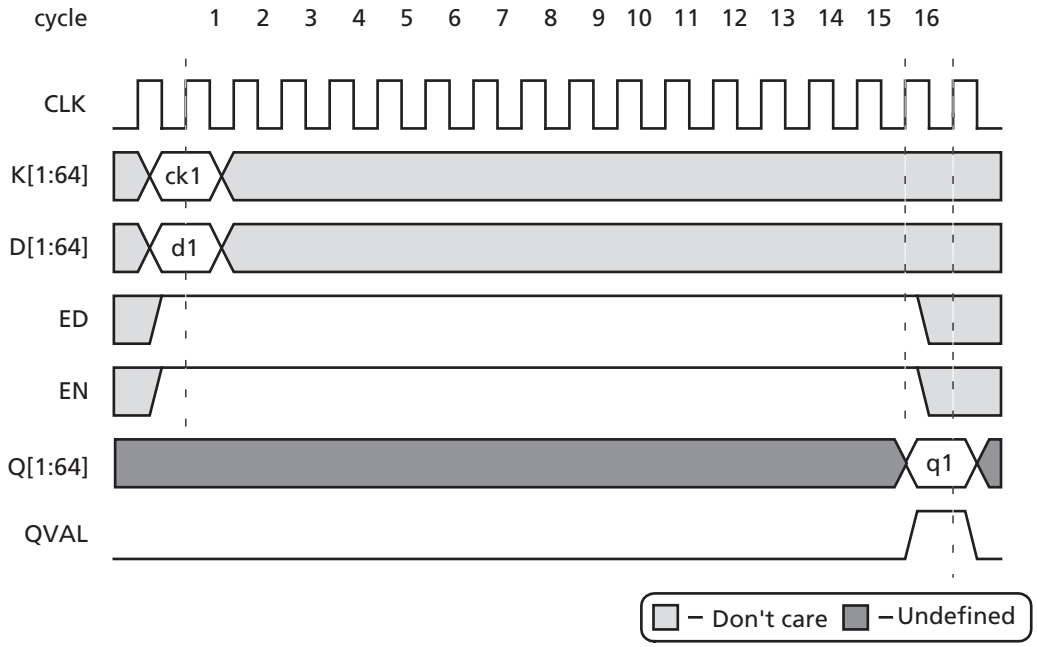

Figure 6 • **Example Encryption Sequence**

Decryption

To begin the process of decrypting data, the following inputs are set:

1. K[1:64] is set to the cipher key (ck1 in Figure 7) to decrypt the data.
2. D[1:64] is set to the ciphertext data (d1 in Figure 7) to be decrypted.
3. ED is set to logic '0'.
4. EN is set to logic '1'.

After 16 clock cycles of the EN input being held continuously at a logic '1' value, the QVAL signal will transition from logic '0' to logic '1' and remain valid for one clock cycle, indicating that valid plaintext (unencrypted data shown as q1 Figure 7) is available on the Q[1:64] outputs.

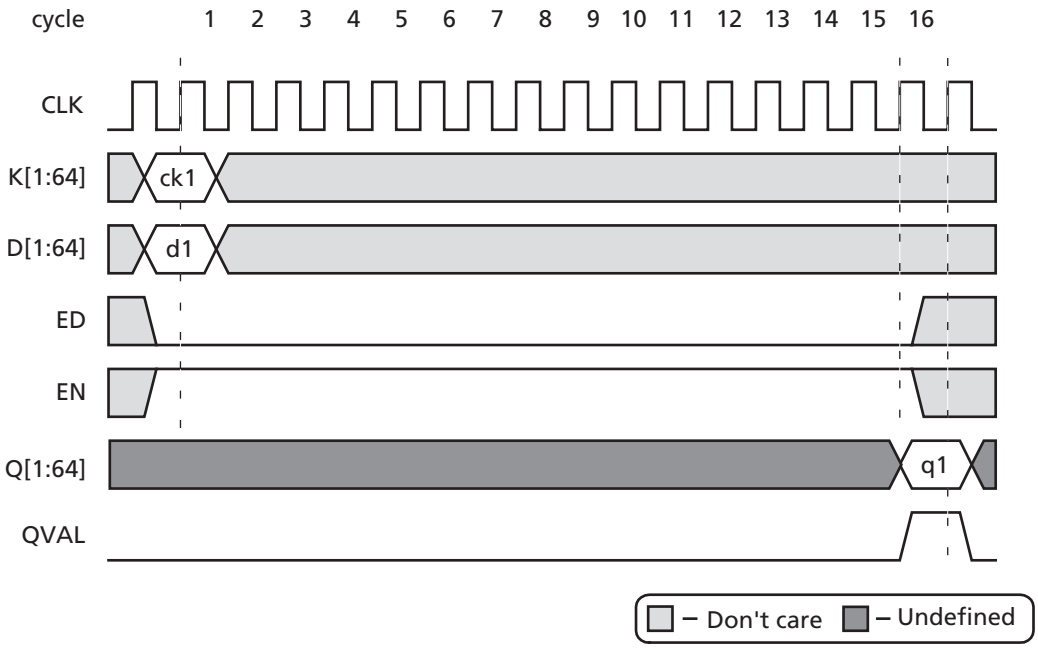

Figure 7 • **Example Decryption Sequence**

Pause/Resume

For normal operation, the EN input is held at a logic '1' value. The core can be paused by holding the EN input at a logic '0' value, indefinitely, as shown in Figure 8. To resume operation, the EN input should be brought back to a logic '1' value. This functionality applies to either encryption or decryption. Note that the ED input must remain at logic '1' throughout an entire encryption cycle, or at logic '0' throughout an entire decryption cycle; otherwise, unpredictable results on the Q[1:64] outputs will occur.

The pause/resume functionality is provided as an aid to the user. One possible use for the pause functionality is the case where many blocks of data are encrypted one after another, for which the EN input can be held statically at a logic '1' value, the data input changing every 16 clock cycles to encrypt the next block. After all blocks of data are encrypted, the user would then need to hold the EN input at a logic '0' value, since if it is left at a logic '1', data will continue to be encrypted ad infinitum. When ready for the next blocks of data, the user can then resume the encryption process by holding the EN input at a logic '1' value. Another possible use may be if the user has an elastic buffer (FIFO) connected to the Q[1:64] outputs. If the FIFO is filling up with encrypted data faster than the encrypted data is being read out of the FIFO, the user may want to pause the CoreDES macro by setting the EN input to a logic '0' when the full or almost-full flag logic from the FIFO is active. When the FIFO full or almost-full flag logic clears, the CoreDES macro can then resume operation by again setting the EN input to a logic '1' value.

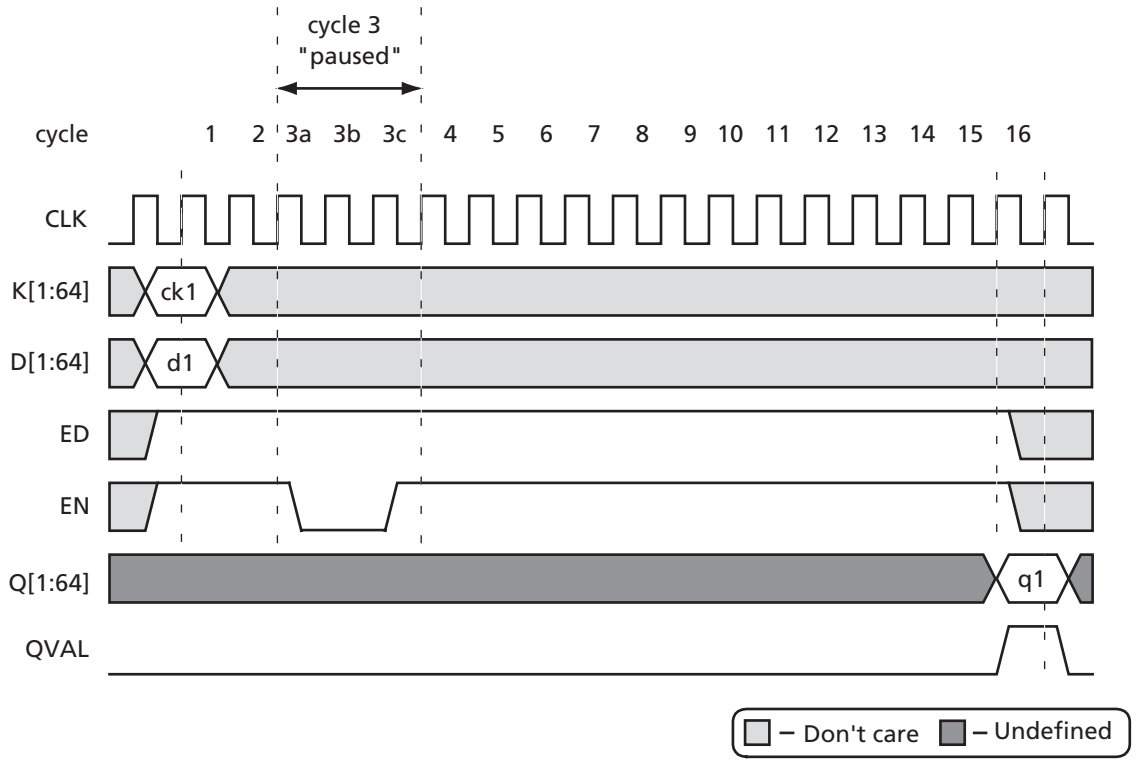

Figure 8 • **Example Encryption Pause/Resume Sequence**

Clear/Abort

At any point in the process of encrypting or decrypting data, the user can abort the current operation by setting the CLR input to logic '1'. This will clear all current calculations with the key schedule and data schedule logic. The user can then immediately begin to use a different cipher key and data input on the very next cycle, as shown in Figure 9.

The clear/abort functionality is provided as another aid to the user. This is employed when the user wants to change the cipher key, possibly in the middle of an encryption or decryption sequence. The user is able to immediately halt the current operation simply by holding the CLR input at a logic '1' value for at least one clock cycle, and commence immediately on the following clock cycle with a new cipher key and/or new data. If the CoreDES macro is integrated into a system containing a processor, the processor may want to abort the encryption or decryption operation for some specific event (e.g., low or failing power condition).

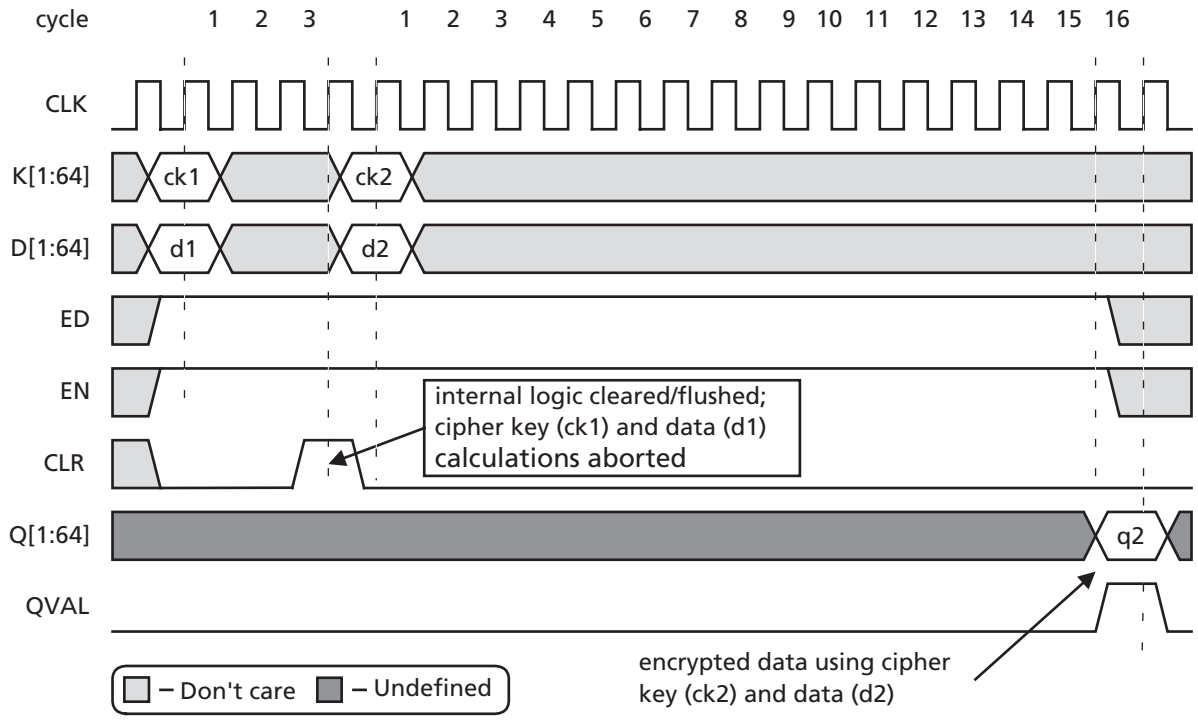

Figure 9 • **Example Encryption Abort Sequence**

Modes of Operation

CoreDES is implemented using the ECB (Electronic Codebook) mode of operation, per *FIPS PUB 81*. Depending upon the application, other modes of operation for DES may be desirable. For this reason, Actel provides example VHDL and Verilog source code for the CBC (Cipher Block Chaining), CFB (Cipher Feedback), and OFB (Output Feedback) modes. For detailed information on specific modes of operation, refer to *FIPS PUB 81*.

Ordering Information

CoreDES can be ordered through your local Actel sales representative. The following number convention should be used when ordering: CoreDES-XX, where XX is listed in Table 3.

Table 3 • **Ordering Codes**

XX	Description
EV	Evaluation Version
SN	Netlist for single-use on Actel devices
AN	Netlist for unlimited use on Actel devices
SR	RTL for single-use on Actel devices
AR	RTL for unlimited use on Actel devices
UR	RTL for unlimited use and not restricted to Actel devices

Export Restrictions

CoreDES is subject to export controls and is licensable under the U.S. Department of Commerce's Export Administration Regulations, the U.S. Department of State's International Traffic in Arms Regulations, or other laws, government regulations, or restrictions. Actel is currently in the process of obtaining additional permissions to ship CoreDES to a wider audience. The licensee will not import, export, reexport, divert, transfer, or disclose CoreDES without complying strictly with the export control laws and all legal requirements in the relevant jurisdictions, including and without limitation, obtaining the prior approval of the U.S. Department of Commerce or the U.S. Department of State, as applicable.

List of Changes

The following table lists critical changes that were made in the current version of the document.

Previous Version	Changes in Current Version (v4.0)	Page
v3.0	The "Supported Families" section was updated to include Fusion.	1
	Table 1 was updated to include Fusion data.	4
v2.0	The "Supported Families" section was updated to include ProASIC3/E.	1
	Table 1 was updated to include ProASIC3/E data.	4
	The "Modes of Operation" section was added.	9

Datasheet Categories

In order to provide the latest information to designers, some datasheets are published before data has been fully characterized. Datasheets are designated as "Product Brief," "Advanced," and "Production." The definitions of these categories are as follows:

Product Brief

The product brief is a summarized version of an advanced or production datasheet containing general product information. This brief summarizes specific device and family information for unreleased products.

Advanced

This datasheet version contains initial estimated information based on simulation, other products, devices, or speed grades. This information can be used as estimates, but not for production.

Unmarked (production)

This datasheet version contains information that is considered to be final.

www.actel.com

Actel Corporation

2061 Stierlin Court
Mountain View, CA
94043-4655 USA
Phone 650.318.4200
Fax 650.318.4600

Actel Europe Ltd.

Dunlop House, Riverside Way
Camberley, Surrey GU15 3YL
United Kingdom
Phone +44 (0) 1276 401 450
Fax +44 (0) 1276 401 490

Actel Japan
www.jp.actel.com

EXOS Ebisu Bldg. 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan
Phone +81.03.3445.7671
Fax +81.03.3445.7668

Actel Hong Kong
www.actel.com.cn

Suite 2114, Two Pacific Place
88 Queensway, Admiralty
Hong Kong
Phone +852 2185 6460
Fax +852 2185 6488

5172172-2/12.05