SANDIA REPORT

SAND2003-3867
Unlimited Release
Printed October 2003

Data Encryption Standard ASIC Design and Development Report

Lyndon G. Pierson, Edward L. Witzke, Perry J. Robertson

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94AL85000.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.doe.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2003-3867
Unlimited Release
Printed October 2003

Data Encryption Standard ASIC Design and Development Report

Lyndon G. Pierson and Edward L. Witzke
Advanced Networking Integration Department

Perry J. Robertson
RF and Opto Microsystems Department

Sandia National Laboratories
P.O. Box 5800
Albuquerque, NM 87185-0806

Abstract

This document describes the design, fabrication, and testing of the SNL Data Encryption Standard (DES) ASIC. This device was fabricated in Sandia's Microelectronics Development Laboratory using 0.6 μm CMOS technology. The SNL DES ASIC was modeled using VHDL, then simulated, and synthesized using Synopsys, Inc. software and finally IC layout was performed using Compass Design Automation's CAE tools. IC testing was performed by Sandia's Microelectronic Validation Department using a HP 82000 computer aided test system. The device is a single integrated circuit, pipelined realization of DES encryption and decryption capable of throughputs greater than 6.5 Gb/s. Several enhancements accommodate ATM or IP network operation and performance scaling. This design is the latest step in the evolution of DES modules.

Acknowledgements

The authors would like to thank the following individuals for their support and assistance in this project. Russell Mikawa and Alan Lundin provided computer-aided engineering and support. Cathy Reber was very helpful in identifying and procuring integrated circuit packages and Brent Meyer's technical advice proved quite valuable. Dave Renninger gave great effort in a difficult reticle design, and all the fabrication personnel of Sandia's Microelectronic Development Laboratory produced parts of high quality. Al Opichka (formerly of Sandia) overcame tremendous hurdles to test the fastest, highest pin-count integrated circuit fabricated to that date, at Sandia. Hans Rodriques de Miranda (formerly of RE/SPEC) performed the initial implementation in programmable logic devices. Karl Gass (formerly of Utah State University) worked on the packaging. Most of all, the authors would like to thank Craig Wilcox (formerly of Sandia National Laboratories), who performed the VHDL design of the SNL DES ASIC, oversaw the fabrication and testing, and provided early drafts of this document.

Contents

Figures

Tables

1.0 Introduction

The Data Encryption Standard (DES) provides a single, standard, cryptographic algorithm for protecting computer and communications information. This standard was first approved by the NBS (National Bureau of Standards now NIST, National Institute of Standards and Technology) in 1977 as FIPS Pub 46 [2]. DES is a Feistel-type block cipher that operates on 64 bit blocks of data using a 56 bit key [2].

Feistel Ciphers [8][11] operate on the left and right halves of a block of bits in multiple rounds. An interesting property of Feistel Ciphers is that the function *f*, used to operate on the half-blocks of data bits, does not need to be invertible for the Feistel Cipher to be invertible. In the Data Encryption Algorithm, the function *f* is a product cipher, because the function performs both substitutions and permutations.

Since 1977, many software and hardware implementations have been designed and fabricated. Previous DES realizations accomplished data throughputs on the order of tens-to-hundreds-of megabits per second. Modern computer networks communicate with data rates in the gigabits per second range and beyond. The advent of massively parallel supercomputers capable of teraflop performance has increased the need for high-throughput encryption. Under the guidance of the Department of Energy's Accelerated Strategic Computing Initiative (ASCI) program, Sandia's Advanced Networking Integration Department has studied methods of scaling encryption to these high-bandwidth requirements. An accessory to these studies was designed and fabricated in a single integrated circuit. Called the SNL DES ASIC (application specific integrated circuit), this device supports high data throughput and has special features to accommodate long-haul data transmissions across asynchronous transfer mode (ATM) or Internet protocol (IP) networks.

The implementation described in this report is a pipelined approach offering encryption, decryption, or algorithm bypassing on a cycle-by-cycle basis. Also on each clock cycle, a unique cryptographic key (for either encryption or decryption) can be input to the ASIC. Eighteen clock cycles process data completely through the pipeline, causing the appropriately decrypted, encrypted, or bypassed data to appear on the ASIC outputs. Additionally, all key and control input signals pass through the pipeline and exit the ASIC synchronized to the ciphertext outputs.

This design was originally implemented as a set of four Altera programmable logic devices (PLDs). (Previous history of the design considerations and performance of the PLD implementation can be found in the paper by Wilcox, et al. [13].) The design was translated into VHDL (VHSIC Hardware Description Language, where VHSIC stands for Very High Speed Integrated Circuit) and synthesized into the Compass library of standard cells. The device was fabricated in the

Microelectronics Development Laboratory (MDL, located at Sandia National Laboratories, Albuquerque, New Mexico) using 0.6-μm CMOS. Two wafer lots were successfully fabricated and produced devices that operate up to frequencies greater than the IC tester limit of 105 MHz.

This device has several enhancements that facilitate encryption for high-speed IP or ATM networks. Individually, this ASIC accommodates throughputs in excess of 6.5 Gb/s. Careful considerations provide multiple device configurations that accommodate 10 to 40 Gb/s throughputs. There are features for indicating start-of-packet/start-of-cell, flagging valid cells, and bypassing cell or packet headers. The device may be cascaded to accomplish triple-DES and there are configurations for single ASIC with external programmable logic to achieve triple-DES. Being a fully static CMOS device, the power usage is proportional to operating frequency. Therefore, in low-bandwidth situations this device draws very little power.

2.0 DES Algorithm Overview

The DES algorithm is a symmetric block cipher. For each plaintext block of 64-bits, a 56-bit key is used to produce a 64-bit block of ciphertext. The same 56-bit key is used to recover plaintext when ciphertext is input. Encryption and decryption use the same key and algorithm, the only difference being the generation of subkeys. DES processes input blocks through permutations, initial and inverse initial, and

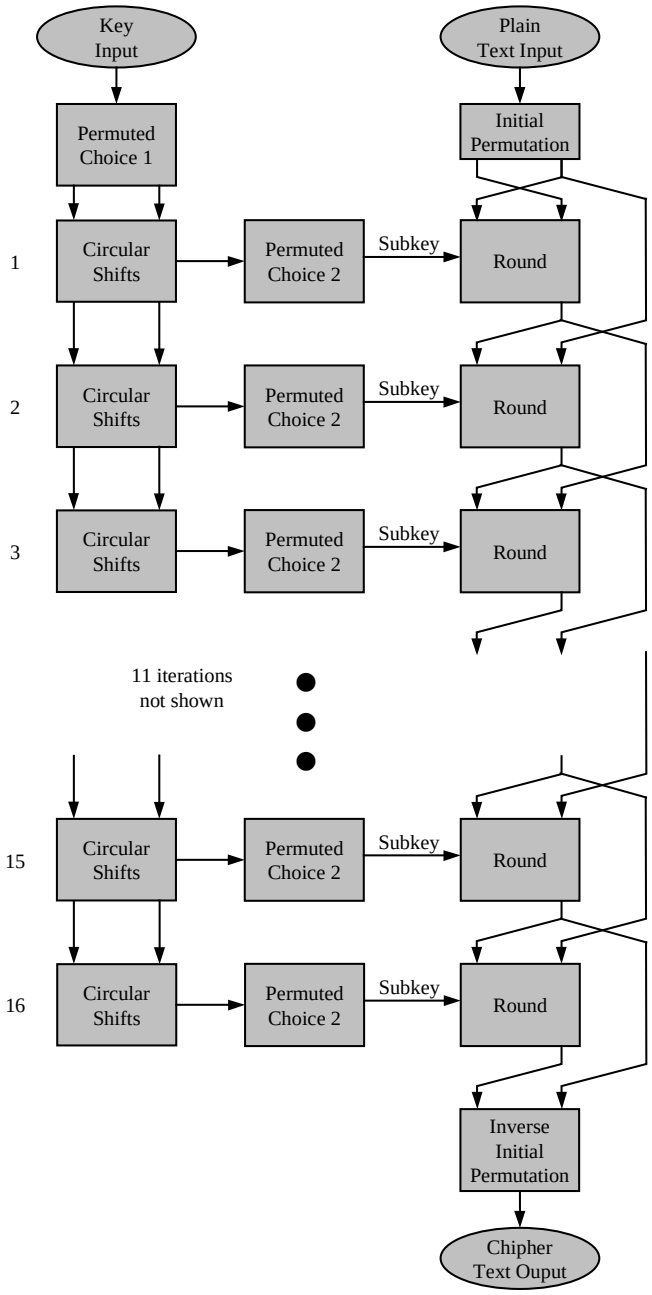


Figure 1. DES Algorithm Block Diagram.

16 equivalent rounds that consist of permutations, summing, and bit-wise manipulations (Figure 1).

The initial permutation simply routes input bits to different locations for processing. For example, the first bit is routed to location 40, and bit 58 is routed to location one. FIPS Pub-46 fully explains this rerouting. The inverse initial permutation merely reverses the initial permutation. Between the initial and inverse initial permutations, the block is processed by 16 rounds as shown in Figure 1. The further details of each round are provided by FIPS Pub-46 [2].

After the initial permutation, the 64-bit block is broken into two 32-bit halves. The right half passes through an expansion permutation. This replicates several bits to expand from 32-bit to 48-bits. FIPS Pub-46 [2] has full details for this expansion and Schneier [11] explains its significance. The right half also exits the round as the left half output.

The output of the expansion permutation is combined with the appropriate subkey in a 48-bit exclusive-or. This output becomes the input for eight s-boxes. Each s-box produces 4-bits of output for 6-bits of input. Specific s-box definitions are given in FIPS Pub-46 [2].

The s-boxes generate 32-bits of outputs that pass to another permutation called the p-box. The p-box simply routes each input to a different location of output as defined in FIPS Pub-46 [2]. A 32-bit exclusive-or combines the p-box output with the left half of the block input. The exclusive-or output then becomes the 32-bit, right half output for the round.

3.0 Preliminary Design and Feasibility Assessment

This development began as a feasibility study for designing and building a single ASIC to perform the DES algorithm at high data rates. Required studies included the algorithm, a previous implementation in Altera PLDs, alternate architectures, potential gate counts, throughputs, and IC (integrated circuit) sizes. Implicit in this study was the development of the DES algorithm as a VHDL model.

3.1 Required Functionality

The proposed device was envisioned as a single IC capable of both the encryption and decryption DES algorithms. A single input bit would select the desired function. This device was designed to allow the clocking in of all input data, including the text data (64 bits) and key (56 bits) and control bits, necessary to perform DES. The data path through the chip was pipelined in order to boost the data throughput. Pipelining implied that all key, text, and control inputs could flow through the IC in parallel so that operations from clock cycle to clock cycle could be distinct. Additional features were a bypass-mode and the synchronization of SOC (Start of Cell – an ATM term) and VALID input bits. Bypass-mode allows text to enter and leave the devices without undergoing encryption or decryption. The SOC and VALID bits were needed to assist ATM and/or IP interfaces. In order to support various modes of encryption, it was necessary that all inputs should traverse the device and appear at the output in sync with the output data (cipher text, key and control). It was required to scale encryption to OC-48 (ATM throughput of 2.5 Gb/s) as a minimum and desired to scale to OC-192 (ATM throughput of 10 Gb/s). To scale ATM encryption to 2.5 Gb/s and beyond, the ASIC was required to operate at a minimum of 40 MHz. For 10 Gb/s, the IC would have to run at least 160 MHz.

3.2 Alternate Architectures

Several alternate architectural styles of the DES algorithm were explored to determine the preferred approach to ASIC implementation. Variations in subkey generation techniques and the use of ROM (read only memory) in place of some combinatorial logic presented options for throughput enhancement. There was also some investigation of full custom IC design methods. Specific techniques included PLA (programmable logic array) implementations or manual synthesis of s-boxes, dynamic logic, and full custom specialized-logic cells. The final implementation benefited from this study, while future efforts may benefit from ideas that were studied but not used.

3.2.1 Different Subkey Generation Methods

There were several apparent ways to generate the subkey for each round of the DES algorithm. The DES algorithm defined subkeys as left circular shifts and permuted choices at each round, resulting in 48 subkey bits used in each round. This resulted in shifted key data progressing through the pipeline. A wire-list method was

defined to pass key data through the pipeline without shifting. For each round, the appropriate 48 bits were selected as determined by the algorithm. Since the ASIC design required the ability to either decrypt or encrypt at each pipeline stage, one of two possible 48-bit subkeys had to be selected for each round. This implied a multiplexer to select the correct subkey as determined by DEN (decrypt or encrypt not) input.

Both subkey generation methods were encoded in VHDL for simulation and synthesis evaluations. With respect to the maximum frequency of operation, the two methods were roughly equivalent. Both implementations required an array of 2-to-1 multiplexers. The shift and permute method required slightly more wiring to allow for the shifting from stage to stage, but neither impacted the pipeline frequency of operation. The prominent difference was the number of logic gates required for the two implementations. The shift and permute method required 56 multiplexers per stage while the wire-lists method needed only 48. The wire-lists method resulted in saving 384 gate equivalents in the ASIC area and proportionate savings in power and reliability.

The possibility of removing subkey generation from the critical path of data encryption or decryption was explored. The idea was to find a method to predetermine the appropriate subkey before the arrival of data at each stage of the pipeline. Each attempt of this yielded the same result, namely, the data, key, and decrypt or encrypt information arrived at the ASIC simultaneously. Therefore, the entire pipeline had to remain synchronized. This prevented any type of look-ahead subkey generation with traditional circuits.

Another subkey generation method was studied and implemented in the DES ASIC. This method utilized the initial permutation of key inputs and wiring from bond-pad cells to the input register. This wiring accomplished both the initial permutation and the first circular shift of key inputs. In order for both decrypt and encrypt modes to work, the input register multiplexed incoming data depending upon the value of the DEN input. Since the properly shifted key data was present on the output of the input register, no multiplexing was needed in the pipeline critical path. For succeeding pipeline stages, multiplexing was accomplished after key data was used for algorithm processing. The appropriate multiplexing was designed to occur at the next register input, thus removing key multiplexers from the critical path and enhancing the overall throughput.

3.2.2 ROM Implementations

Four alternate implementations using ROMs were investigated for potential enhancements of ASIC size and speed. The intent was to use ROMs to implement some of the combinatorial logic in each round, and compare the size and operational frequency effects on the ASIC. The four proposed architectures were 512 each, 6x1 ROMs, 128 each, 6x4 ROMs, 512 each, 13x1 ROMs, and 128 each, 16x4 ROMs.

In the first case, 512 ROMs each with 6 address inputs and 1 output were to replace the s-boxes in each round. For 16 rounds with 32 s-box outputs each, this resulted in 512 total ROMs. This architecture violated the minimum constraints in the ASIC ROM compiler from Compass software used at the time. The minimum output width for this compiler was four bits resulting in inefficiency of four copies of address decode repeated in the four ROMs used to implement one s-box.

Another architecture replaced each s-box with one ROM having six address inputs and four output bits. For the entire algorithm, this required 128 ROMs, eight per stage, for each of 16 pipeline stages. This matched the minimum size of the ROM compiler. Comparing the total area to the one needed for a gate-level implementation, this architecture was 50 percent larger. The ROM's minimum cycle time of 4.3 ns however, was faster than gate-level synthesis. This would have reduced the nominal clock period to 6.6 ns or an operating frequency of 152 MHz and throughput of 9.69 Gb/s.

Two other ROM architectures, using 512 each 13x1 ROMs and 128 each 16x4 ROMs, were evaluated. Both of these topologies replaced exclusive-ors and s-boxes in the algorithm. These resulted in areas between seven and ten times larger than the area of a gate-level implementation. Additionally, the ROMs in these cases would have been slower than the Synopsys synthesized gate model.

In conclusion, only the 6x4 ROM architecture offered any advantage over the gate implementation, a 1.9 ns speed improvement at the cost of a 50% area increase. In pragmatic terms, this speed gain was offset by the increase in area, the associated increase in wiring, the parasitic delays of extra wiring, an increased margin for error in ROM programming, and greater complexity of the design. The main reason for avoiding the ROM implementations was the ROM compiler itself. This tool had not been used, and generated only synchronous ROMs. Synchronous ROMs needed clocks to operate and would have added pipeline latency. Additional pipeline latency coupled with the unknown performance of the compiler forced the decision of a gate level implementation.

3.2.3 Other Design Approaches

Several other design approaches were considered. Implementation of the s-boxes by full custom specialized-cells, PLA (programmable logic array), dynamic logic, and manual synthesis based on Karnaugh Maps were explored. These methods would most likely have reduced the active area and increased the throughput. However, such manual intervention would have increased cost, design duration, and the probability of human error. Dynamic logic implementations were discarded because they would require rework to allow any future radiation-hardened implementation. Dynamic logic also imposed minimum frequencies of operation that would eliminate low-speed and low-power applications. For the development of a research, proof-of-

concept ASIC, these risks were deemed inappropriate and excessively expensive. Therefore, the decision was made to synthesize a gate-level implementation in static CMOS.

3.3 Synthesis Estimates of Size and Speed

Synopsys Design Compiler software was used to explore the design space required to implement this algorithm. Design Compiler transformed VHDL constructs into library-cell gate implementations of the target function. Size and speed boundaries were derived from user input constraints. This software allowed the evaluation of tradeoffs between size and speed for various optimizations. Listed here (in Table 1) are several combinations of equivalent gate areas and the associated operational frequency. These were extrapolated from the synthesis of a single pipeline stage. The data represent only the internal frequency, not the estimated actual frequency of the proposed ASIC. Actual frequencies would be somewhat reduced due to the input and output delays inherent to on-chip and off-chip interfaces.

Table 1. Preliminary Synthesis Results

Equivalent Gate Area	Pipeline Delay (ns)	Frequency (MHz)	Throughput (Gb/s)
43296	14.11	70.9	4.54
43766	13.48	74.2	4.75
43550	12.10	82.6	5.29
55532	8.62	116.0	7.42
56206	8.67	115.3	7.38
56543	8.55	117.0	7.49
57575	8.51	117.5	7.52

3.4 Comparison of SNL DES to NSA DES Synthesis

Mark Bean of NSA provided a copy of his paper, "Evaluation of Hardware Implementations of the DES Algorithm." This paper [1] had many similarities to the DES ASIC described here. The NSA design was a 16-stage pipeline that allowed encrypting or decrypting, 64 bit data and 56 bit keys, but they did not mention a bypass option. This design was synthesized using Synopsys software as well. A primary difference was the target cell library. NSA targeted a 0.8-µm CMOS compared to our Compass Passport library of 0.6-µm CMOS. By comparison, the Compass library would have been smaller, faster, and lower power. Cell availability differences between the two libraries caused some variance in gate requirements. There is more variety in the Compass library that gave the synthesizer more latitude to implement logic functions. The particular VHDL constructs used to model the algorithm imply various logic implementations. This means that various VHDL models of the same algorithm synthesized differently.

NSA reported an operational frequency of 62.9 MHz for a throughput of 4.03 Gb/s. This design occupied 262,000 transistors. The standard conversion of transistors to CMOS gates is to divide the number of transistors by four, the number of transistors in a two-input nand gate. This yields 65500 gates. They did not report the equivalent gate area or measure of square microns to implement this in 0.8-µm CMOS. Also, NSA proposed using half-as-many multiplexed inputs and half-as-many multiplexed outputs to reduce the number of IC interface pins. This required a separate clock running twice the frequency of the internal clock.

This synthesis compared quit well with preliminary analyses and had similar design space bounds. The differences could be easily explained by variations in the cell libraries and VHDL coding styles. NSA's operational frequency, 62.9 MHz, was expected to be a bit slower because of the use of in 0.8-µm CMOS. With similar gate counts and estimated throughputs, the NSA evaluation served as a rational sanity check and valid evidence to support the DES ASIC development.

3.5 Preliminary Assessment Conclusions

The preliminary assessment indicated a feasible ASIC design implementation of the DES algorithm that would make an ideal proof-of-concept vehicle for prototyping high-speed encryption and decryption. The implementation successfully encrypted and decrypted the test vectors given in FIPS Publication-81 [4], showing correctness in the VHDL model of the algorithm. Synthesis of the design placed it in the "ball-park" of NSA's DES evaluation with understandable differences. This synthesis yielded a design that physically fits IC fabrication limits with sufficient margin. The synthesis projections on operational frequency indicated sufficient margin to achieve 2.5 Gb/s throughput and, perhaps, well into the gray region between 2.5 and 10 Gb/s throughput. These results led to funding for the development of the DES ASIC.

4.0 Functional Design

Once a DES ASIC appeared feasible, the ASCI program funded the design efforts. Previous Laboratory Directed Research and Development (LDRD) efforts had completed a PLD implementation that became the foundation, upon which to build. The feasibility study and early work in the ASIC design revolved around learning the LDRD efforts, understanding the algorithm, defining design requirements, converting the design into behavioral VHDL, and assuring accurate algorithm modeling. Accuracy was determined by comparison of test bench simulations to encryption and decryption test vectors published in FIPS Publication 81 [4]. The functional design was completed by converting from behavioral VHDL to RTL (register transfer level) VHDL and simulating against the previous test benches.

4.1 Overview of Encryption Engine Design

The following sections contain the rationales for some of the design decisions regarding various features of the SNL DES ASIC.

4.1.1 Bypass

For end-to-end encryption applications, the network routing information in the header of a packet or cell must be sent unencrypted in plaintext form. Therefore a method of bypassing the encryption or decryption function for each input word would eliminate the need to design a separate data path around the encryption chip. Enabling a bypass mode within the encryption chip also removes the difficulties of synchronizing the plaintext data passed around the chip with the encrypted/decrypted data passed through the chip.

The "Bypass" input pin is used to signal to each pipeline stage that a null encryption (pass-through) of data is to be performed for the associated input word as it passes through each stage. Together the Bypass and the Valid/SOC bits signal to each pipelined stage whether the data entering that stage is to be bypassed, processed with the pipelined key, or ignored as if it were an "idle" or "unused" bit (such as a word of bits between valid cells or packets).

4.1.2 Valid and SOC

The "Valid" input pin is used to flag the "idle" or "non-valid" words being processed by the pipeline. This bit can also be used to mark the end of packet for variable packet length applications. This bit is simply a flag that accompanies each word that enters the pipeline that can be used to mark words entering the pipeline so that the same words can be easily identified when they exit the pipeline. Without this bit, the identification of words output from the pipeline that are valid words of a communicated packet or cell would be more difficult and would require additional external circuitry.

The Start of Cell or “SOC” bit is also a flag that accompanies the data words in the encrypt/decrypt pipeline. This flag signals if the accompanying block of data is the start of an ATM cell (or the start of a variable length packet). If the SNL DES ASIC is to be used for a purpose other than encrypting ATM cells, this bit could be redefined as appropriate to provide any user-defined information (e.g. start-of-packet for an IP network) to travel with the input data and key.

In a future version of this ASIC, the design may be modified to incorporate additional input bits that may be associated with the input word as it passes through the pipeline stages. In a design that allows output words to be interleaved back into the input stream for "multiple encryption" operation, these additional bits would be useful to mark individual words that have passed through the encryption operation a second and/or third time (e.g., for Triple DES sharing the bandwidth of a single 16-round pipeline).

4.1.3 Both Encrypt and Decrypt

Because DES is a Feistel Cipher, the same multiple round process is used for encryption and decryption, but the subkeys are used in different order. Therefore, an encrypt/decrypt or “E/D” input pin is used to select whether encryption or decryption is to be applied to the input word as it passes through each stage of the pipeline. Specifically, this encrypt/decrypt bit selects the proper subkey for either encryption or decryption for each pipelined stage. A "one" input to this pin selects encryption, and a "zero" selects decryption. Note that encryption or decryption can be specified on a "per input word" basis, allowing encryption operations to be fully interleaved with decryption operations.

A useful re-design of the processing of this bit would be to provide an inverted output for coupling into a cascaded encryption chip. Triple-DES and other cascaded uses of the DES usually involve full 16-round encryption in one stage followed by decryption in the second stage and then encryption in the third stage (E-D-E operation). If three DES ASICs were cascaded to process triple-DES, an inversion of the E/D bit from one cascaded stage to the next would facilitate the "E-D-E" mode without use of an external inverter, as described and illustrated later in Section 8.3.

4.1.4 All Key Bits Output

In addition to the plaintext/ciphertext words that normally exit an encryption ASIC, the pipelined Key words, the Bypass, Valid, SOC, and E/D bits are also output from the ASIC in order to cascade multiple devices to implement Triple DES or other extensions of the DES algorithm. Normally, the Key bits would not be output from the encryption chip, as this might provide another avenue by which the adversary might gain access to the cryptovariables. In a future version of this ASIC, three pipelined 16-round sections (for a total of 48 rounds for fully pipelined Triple-DES) might be incorporated into a single ASIC, eliminating the need to output these pins for cascaded operation. This would not by itself reduce the pin count, however,

since additional pins would be required to input the multiple keys needed for "Two-Key" or "Three-Key" Triple DES operation.

4.2 Design Requirements

In order to be useful as a research device, the DES ASIC had to be designed to meet certain minimum requirements of functionality. Specifically, these minimum requirements were:

- The ASIC must provide DES encryption as defined in FIPS PUB 46.
- The ASIC must provide DES decryption as defined in FIPS PUB 46.
- The ASIC must correctly encrypt and decrypt the sample blocks in FIPS PUB 81, Appendix A, Table B1.
- The ASIC must provide compatibility with the DES modes of operation defined in FIPS PUB 81.
- The ASIC must provide a pipelined architecture to maximize data throughput.
- The ASIC must provide a bypass mode to allow the flow of data without perturbation by encryption or decryption.
- The ASIC must provide pipeline synchronization for the ATM signals SOC and VALID.
- The ASIC must facilitate triple DES encryption and decryption by providing pipeline-synchronized inputs and outputs for 56 bit keys and 64 bit texts (plain or cipher).
- The ASIC must provide clock cycle agility. Specifically, it must accommodate encryption or decryption with a new key input for each clock cycle.
- The ASIC must meet OC-48 throughput of 2.5 Gb/s (operational frequency of 40 MHz as a minimum). It is desired for the ASIC meet OC-192 throughput of 10 Gb/s (operational frequency of 160 MHz as a minimum).
- The ASIC may have additional features for Ping-Pong mode. This would allow two ASICs running in parallel and using opposite clock phases to achieve OC-192 while individually operating at 80 MHz clock frequency.
- The ASIC must operate in a nominal 5 Volt, commercial environment.

4.3 Behavioral VHDL Design

In order to use modern IC design tools, it was necessary to model the DES algorithm in VHDL. VHDL provided a platform for standard modeling of electronic components and systems. VHDL also allowed for a simulation model very similar to a software implementation. Divorced of the burden of parasitic delays and the physical attributes of actual electronics, VHDL represented a fast, time efficient

method of developing and verifying the operation of the DES algorithm. The VHDL modeling, simulation, and synthesis was performed using Synopsys, Inc. software.

The behavioral VHDL model was developed from a previous Sandia implementation of the DES algorithm. Earlier designs started with a spreadsheet version of the algorithm. This, in turn, was captured as an Altera PLD design [13]. The Altera design consisted of schematics and text files written in AHDL (Altera Hardware Description Language). The Altera design had been simulated for comparison to the spreadsheet operation and tested by means of back-to-back inverse operation. Back-to-back inverse operation verified that two devices cascaded together would produce the original input if the two devices ran opposite functions (encrypt-then-decrypt or decrypt-then-encrypt). Spreadsheet and Altera simulations provided a strong sense of correctness in their associated models. These became the initial standards of comparison for the early versions of behavioral VHDL models.

Two small problems arose when the VHDL model was compared to FIPS Pub-46 [2]. These were first discovered in failed attempts to decrypt ciphertext using the model. The Altera design did not contain an initial permutation or inverse initial permutation. These permutations were described in FIPS Pub-46 and were necessary for complete DES compatibility. FIPS Pub-46 described the DES by numbering inputs and outputs as increasing from left to right. Digital logic design has traditionally numbered bits as decreasing from left to right, thus following significance of bits. Both the Altera design and the early VHDL models followed the format of traditional digital logic. This created problems in s-box definitions and trouble-shooting comparisons between the VHDL model and the algorithm description. To achieve the first working model of DES VHDL, modifications of the numbering of bits were required in every design file. After numbering modifications and the inclusion of initial and inverse initial permutations, the VHDL correctly encrypted and decrypted text inputs. The initial model included a bypass mode, but no accommodations for SOC and VALID signals. These signals were added when the behavioral VHDL was converted to RTL VHDL.

4.4 VHDL Test Benches

Another benefit of VHDL modeling came from the use of VHDL test benches. By definition, VHDL was designed to use the same syntax and constructs for simulation stimulus as for hardware modeling. For the DES ASIC, test benches were created to verify functionality, maintain conformance between versions of the design, validate sub-functions, provide self-checks after modifications, generate test patterns for hardware verification, and create stimulus and expected responses for post-layout timing simulations. After the design completion, these same test benches provided foundations for building simulations of triple DES implementations and multiple chip configurations for high-throughput.

A total of five test benches were developed for the DES ASIC. All of these used stimulus contained in external text files and wrote responses to external text files. This facilitated the creation of test vectors for IC testing. Common to these test benches was the clock stimulus generation. Each file included a constant named 'period' that was used to define the clock frequency. The simulations ran one clock period at time. During each period, the test benches followed a recursive process:

1. Read stimulus from a predefined external stimulus file.
2. Force inputs to values defined by this stimulus.
3. Wait one-quarter of a clock-period.
4. Force the clock input 'high.'
5. Wait one-half of a clock-period.
6. Force the clock input low.
7. Wait four-tenths of a clock-period.
8. Read values of device outputs.
9. Write input stimulus and output responses to an external response file.
10. Wait one-tenth of a clock-period.
11. Repeat until the input stimulus file is exhausted.

The self-check test bench was used to verify algorithm functionality. This was the only test bench that deviated from the common process. Process deviation consisted of checking responses against expected responses then reporting errors to the simulation log file. The stimulus file for this test bench incorporated the sample blocks defined in FIPS Pub-81 and examples that were verified by the original spreadsheet.

The generate-random test bench used random number generators to create arbitrary stimulus for text and key inputs. In this test bench DEN, BYPASS, VALID, and SOC inputs were stimulated as periodic signals. VALID and SOC were stimulated to opposite states and alternated every clock cycle. DEN and BYPASS were arranged to pass through all four possible combinations every four clock cycles. During the period of four cycles the key and text inputs were held constant at arbitrary values created by random number generators. This stressed the model by changing modes on every cycle so that each pair of key and text inputs was processed as encrypt, bypass-encrypt, decrypt, bypass-decrypt. Sixteen thousand vectors were created with this test bench. Twenty of these vectors were spot-checked against the spreadsheet.

The generate-test-vectors test bench was used to create test patterns (vectors) for IC acceptance testing. By commenting and uncommenting appropriate lines of code, this test bench was able to create three distinct test vector files. In the functional mode, this test bench included the random vectors created by the generate-random test bench and added vectors to cover the samples in FIPS-81. It also added vectors to cover the 'corner' operations of all '1's, all '0's, and alternating '01' patterns. The

tristate mode was used to test 0-to-tristate, 1-to-tristate, tristate-to-0, and tristate-to-1 conditions. The parametric mode tested 0-to-1 and 1-to-0 output modes.

The ping pong test bench was used to test the operation of two DES ASICs running in parallel using opposite phases of the clock. This test bench used the functional stimulus from the generate-test-vectors test bench.

The back-to-back test bench was used to test the encrypt-decrypt and decrypt-encrypt modes of two DES ASICs cascaded together. It also included a 64-bit wide, 36 stage shift register (to match twice the 18 stages for back-to-back operation) for shifting text in parallel with DES ASIC processing. This test bench checked for differences between the processed data and the data that propagated through the shift register. For input stimulus, the generate-test-vectors test bench vectors were used.

5.0 ASIC Design

After implementing the DES algorithm in the set of four PLDs, the design was translated into VHDL and synthesized into the Compass library of standard cells so that it could be fabricated in Sandia's fab. The device die (Figure 2) was 11.1 mm on each side and was fabricated in Sandia's MDL (Microelectronics Development Laboratory) using the CMOS VI process. The design rules were 0.6 um. Two wafer lots were successfully fabricated and produced devices that operate up to frequencies greater than the IC tester limit of 105 MHz.

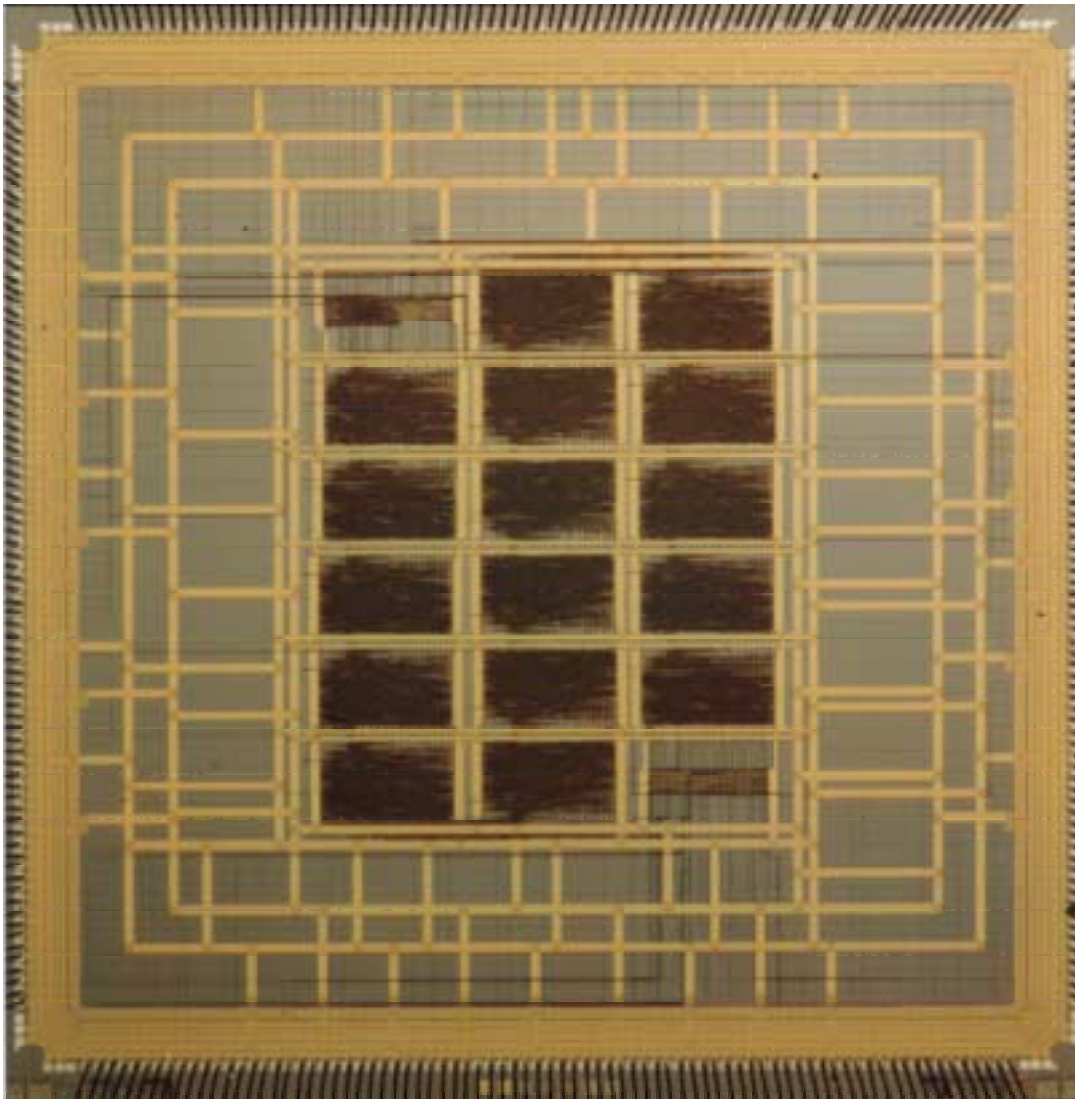


Figure 2. DES ASIC Die.

5.1 Pipelining

This implementation is a pipelined approach offering encryption, decryption, unique cryptographic key input, or algorithm bypassing on each clock cycle. Eighteen clock

cycles process data completely through the pipeline causing the appropriately decrypted, encrypted, or bypassed data to appear on the ASIC outputs. Additionally, all key and control input signals pass through the pipeline and exit the ASIC synchronized to the ciphertext outputs.

At the time of its development, the SNL DES ASIC was the only known fully pipelined implementation of all 16 rounds of the DES algorithm. Pipelining increased the device throughput by dividing the algorithm into equally sized blocks and latching information at the block boundaries. This gives signals just enough time to process through each block between clock cycles, thereby maximizing the operational frequency.

5.2 Key Agility

Pipelining the algorithm allows a high degree of key and function agility for this device. Here, agility means that the SNL DES ASIC can process inputs differently on each clock cycle. For each block of data (on every tick of the clock), the SNL DES ASIC can change keys and select encryption, decryption, or bypass as the mode in which to operate on the data. As an example, the device may encrypt data with one key on one clock cycle, decrypt new input data with a different key on the very next clock cycle, bypass the algorithm (pass the data unencrypted) on the following clock, then encrypt data with yet another independent key on the fourth clock cycle. High speed, key-agile encryption allows each user on a workstation or server to use different key material. Each session through a common network interface may utilize a separate key. This cryptographically separates the users' traffic. This key agility was designed to enable the separate encryption of many virtual channels within a high-speed communication system. The control signals used to select these various modes of operation are presented at the output, passing through the device synchronized to the input data and the input key information. All inputs and outputs (control, key, and data) enter and exit the part synchronously. No other part is known that outputs the key synchronized with the data.

The SNL DES ASIC is the only DES chip known to be able to encrypt or decrypt with a new key, or pass information unprocessed, on each and every clock cycle. This not only allows separate cryptographic processing of many data channels, but additionally enables cryptosystems to be built with fewer components and lower cost. Also, as stated previously, no other encryption chip outputs the key corresponding to each data word in every clock cycle. This per-cycle input and output of all variables facilitates cascading the devices for increased encryption strength, and paralleling the devices for even higher throughput.

5.3 Fabrication Technology

The SNL DES ASIC was fabricated with static 0.6 micron CMOS technology. Its die size is 11.1 millimeters square, and contains 319 total pins (251 signals and 68

power/ground pins). All outputs are tristate CMOS drivers to facilitate common busses driven by several devices.

5.4 Inputs and Outputs

The SNL DES ASIC accommodates the full input of plain or cipher text, 64 bits, and a complete DES key of 56 bits. (Any stripping of unused or parity bits from 64 bit DES keys to achieve a 56 bit key must be performed off chip.) Additionally, 120 synchronous output signals provide 64 bits of cipher or decrypted plain text and the 56 bit key.

Three input only signals control electrical functions for logic clocking (CLK), logic reset (RST), and the tristate output enables (OE). The CLK signal provides synchronous operation and pipeline latching on the rising edge. Both RST and OE are asynchronous, active high signals.

Two synchronous signals, DEN and BYP, determine the DES cryptographic functionality. On the rising edge of each CLK, the logic value presented to the DEN input selects whether input data will be encrypted (logic 1) or decrypted (logic 0). In a similar manner, BYP selects algorithm bypassing (logic 1) or not (logic 0) for each clock cycle. Both of these signals pipeline through the ASIC and exit the device synchronous with the key and data.

Another unique feature of this device is its ability to pass two user-defined control bits in synchronism with the data being encrypted, decrypted, or bypassed. Although they are merely data bits that may provide any user-defined information to travel with input text and key, for this implementation we have defined them as SOC and VAL, for start-of-cell and valid. This capability is indispensable for the design of ATM data encryptors and also for IP encryptors, which must identify the Start of Cell or start of packet boundaries and for systems that must flag data as “valid” or “not valid” in the encryption/decryption pipeline.

6.0 ASIC Performance

6.1 Test Results

ASICs from two wafer lots operate beyond the maximum frequency (105 MHz) of Sandia's IC Test systems. For 64-bit words, this equates to 6.7 Gb/s. (Simulations predict proper operation up to 9.28 Gbps.) This operational frequency was tested over a voltage range of 4.5 to 5.5 Volts and a temperature range of –55 to 125 degrees C.

6.2 Power consumption

Being a fully static, synchronous, pipelined CMOS design, the power usage is proportional to operating frequency. At 105 MHz, the SNL DES ASIC consumes 6.5 Watts of power. While designed to dissipate the heat generated in high-bandwidth applications, the SNL DES ASIC can be operated at much lower data clock rates, consuming very little power, thus enabling many low speed, extremely low power applications. For example, as shown in Table 2, operating from a 3V power supply, the DES ASIC uses a mere 54 mW at 1 MHz, producing a throughput of 64 Mbits per second.

Table 2. Power Consumption of the SNL DES ASIC

	0.01 MHz	**1 MHz**	**105 MHz**
3.0 Vdd	510 μW	54 mW	*
3.3 Vdd	*	66 mW	*
5.0 Vdd	*	165 mW	6.5 W

*untested

6.3 Improving Performance

Since the first SNL DES ASIC was fabricated, several performance enhancements have been identified. These enhancements would increase throughput, aid in cascading devices, and ease the use at the board level. Projected performance enhancements include using improved design tools, improved synthesis options, low-voltage high-speed I/O buffers, and processing in higher-speed technologies.

Several design techniques could improve the design of the existing SNL DES ASIC. For example, recent synthesis developments would allow the DES ASIC to be redesigned with more pipeline stages. A greater number of balanced pipeline stages would increase the operational frequency and boost the throughput beyond 10 Gb/s. To enhance the high-frequency operation at the circuit-board level, different input-output buffers would reduce noise. The present design uses CMOS level (0 – 5 V) interfaces. Future designs would incorporate low voltage transitions to reduce power and noise. Bringing out the clock with the data (source synchronous clocking [14]) would facilitate higher speeds and greater performance. Also, optionally

inverting the encrypt/decrypt output would better facilitate encrypt-decrypt-encrypt triple DES (described in Section 8.2.1).

Moving to a state-of-the-art commercial CMOS process should yield a factor of 4 improvement in the speed of the ASIC. This would produce expected throughputs of about 40 Gbps.

To achieve higher total throughput, multiple SNL DES ASICs can operate in parallel, with each ASIC processing a 64 bit block of the data stream. Figure 3 contains an example of multiple devices, performing DES operations on two blocks of data in parallel.

Because the data outputs of the SNL DES ASIC are tri-stated, there are several ways the ASICs can be used in parallel. The data outputs from both ASICs can be connected to a single output bus in a time-multiplexed fashion and the two DES ASICs can be operated using opposite clock phases to double the data throughput to greater than 13 Gbps. If both ASICs are driven off the same clock edge, the two 64 bit wide data outputs can also be combined into a single 128 bit wide output to achieve the 13 Gbps throughput.

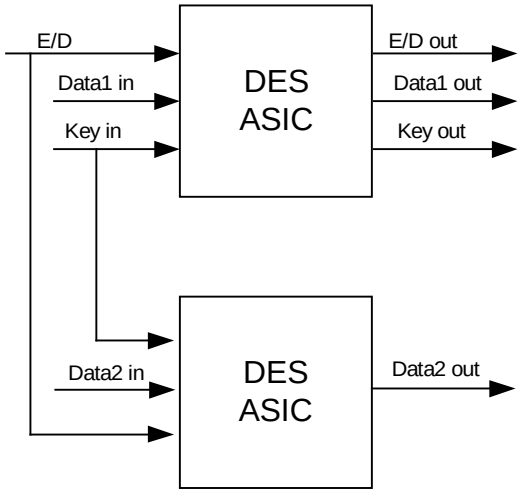


Figure 3. Parallel DES ASIC Implementation.

For encryption of Asynchronous Transfer Mode (ATM) communication sessions where six SNL DES ASICs could operate in parallel on 64 bit blocks to encrypt a 384 bit payload, 40 Gbs (OC-768) rates could be achieved. The authors would expect six parallel DES ASICs made using a state-of-the-art commercial CMOS process to support encryption at 160 Gbps and beyond.

7.0 Package Development

The SNL DES ASIC has been packaged into three different packages including a 360 pin PGA, a 503-pin PGA and a 352 pin BGA. The original 360-pin package was used in initial testing of the DES ASIC performance. It was in this package that the DES chip was shown to operate at over 105 MHz (6.72 Gbps).

Sandia had earlier developed a 1.1 million gate PLD board that used 11 Altera 10K100 devices[10]. This board was used in the development of the DES ASIC pipeline design, housing the four 10K100 devices. It was determined that the SNL DES ASIC could be used with the original PLD11 board, being substituted for a single 10K100 device, if a 503-pin equivalent package were available. Sandia designed an FR4 board onto which the DES ASIC was wire bonded and 503 pins could be inserted. The chip-on-board package had to be designed to dissipate up to 5 watts produced by the DES ASIC. This is accomplished by attaching the DES die directly onto a gold plated copper insert that is attached to the FR4 board using a tin-lead solder preform. Pictures of this package and representative cross section are shown in Figures 4 and 5.

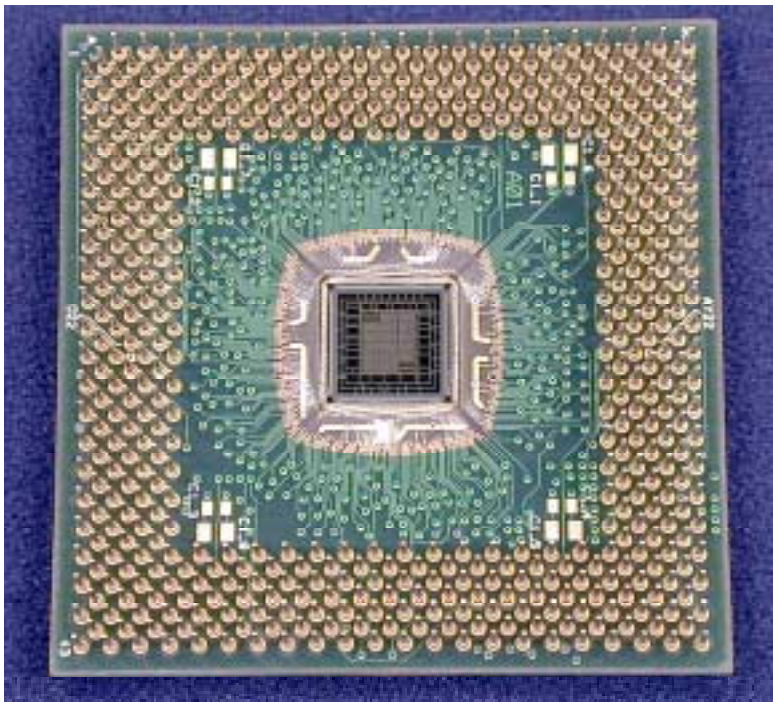


Figure 4. The 503-pin FR4 Board Package.

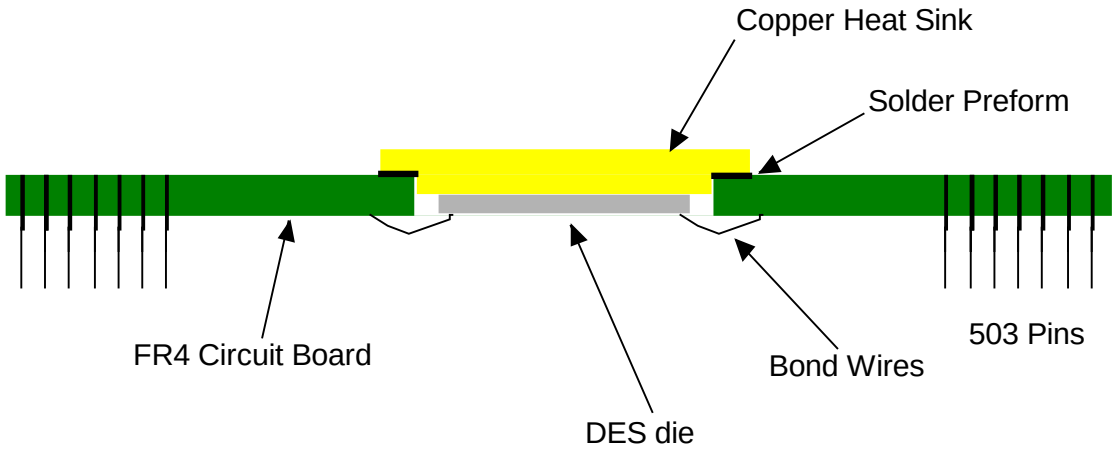


Figure 5. Cross Section of the 503-Pin Package.

The design of this package enables a heat sink and integrated fan to be attached to the back of the copper insert to enable the package to dissipate up to 5 watts. The FR4 printed wiring board uses 3 mil copper traces and spaces with 5 mil vias. This design also allowed the board to be used to connect the existing bus signal assignments from the PLD11 board to the appropriate key and text signals on the SNL DES ASIC. Two versions of the package were designed and fabricated. Each has a different wiring schematic designed to fit into a different socket on the PLD11 board. SNL DES ASICs in the 503 pin package were demonstrated at the Super Computing 98 Conference (SC'98) in Orlando, Florida in November 1998, operating over a dynamic range from 27 bits per second through several hundred megabits per second. Subsequently, in the communications lab, these DES ASICs in the 503 pin packages were operated at multi-gigabit rates on the PLD11 board.

The SNL DES ASIC has also been packaged in a 352-pin ball grid array (BGA) package, shown in Figure 6. This is a commercial package, 35x35 mm from Abpac Inc. of Phoenix, Arizona. The package was chosen for its capability to dissipate over 5 watts, its small size, and its low cost. Abpac's automated manufacturing capability enabled a reduction in packaging costs by more than a factor of 20.

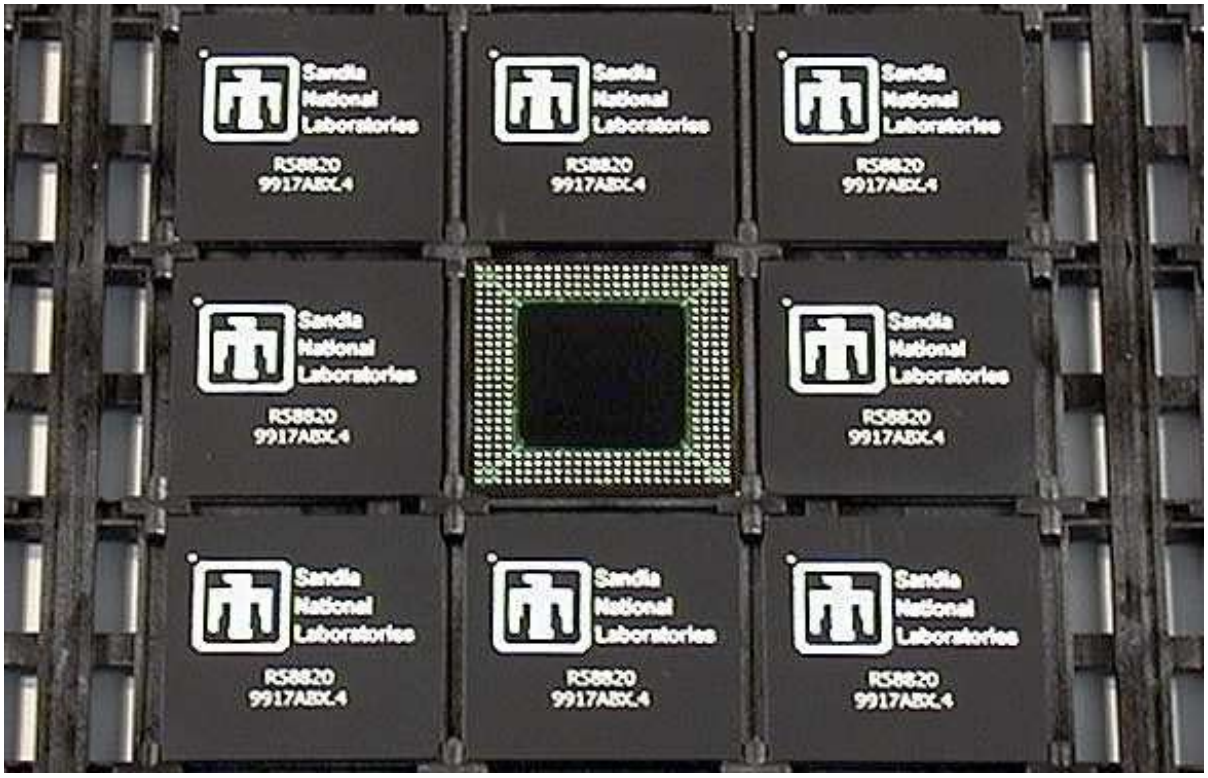


Figure 6. SNL DES ASIC Packaged as a 352 Pin Ball Grid Array.

8.0 Applications of the SNL DES ASIC

The SNL DES ASIC is a versatile device. This chip is a DES engine, and can support various modes and options as described below. Being purely a DES engine, these options, modes, and other support such as key management, are handled off chip by "glue logic." This glue logic can be developed in programmable logic devices (PLDs) or as separate ASICs. Any future versions of the DES ASIC may incorporate portions of this glue logic to provide enhanced functionality on chip, within the DES ASIC

8.1 Modes of Operation

The SNL DES ASIC, as strictly a DES engine, basically operates in Electronic Codebook (ECB) Mode. With the appropriate glue logic, the SNL DES ASIC can also operate in Cipher Block Chaining (CBC) Mode, Cipher Feedback (CFB) Mode, and Output Feedback (OFB) Mode as described in FIPS Publication 81 [4]. Again, with the appropriate glue logic, the SNL DES ASIC can support other modes such as Plaintext Feedback (PFB) Mode and Counter Mode. Examples of the DES ASIC and the surrounding glue logic for CBC Mode encryption and decryption are shown in Figures 7a and 7b. Through proper design, this glue logic can be combined, so one DES ASIC could perform both encryption and decryption.

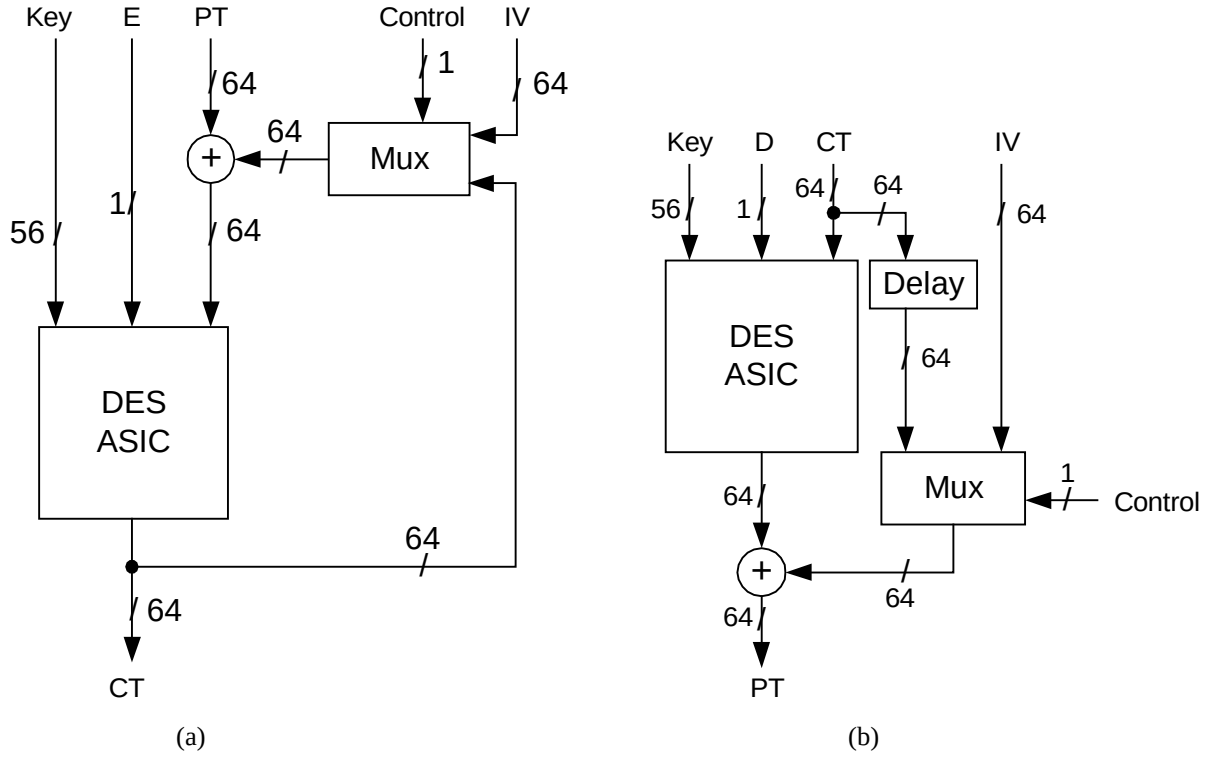


Figure 7. CBC Mode Encryption (a) and Decryption (b).

Although mainly used as an encryption engine for single DES or triple DES cryptosystems, the SNL DES ASIC has other uses. It can be used as a data randomizer. Some encryption algorithms need to hide or obscure relationships between bits or bytes of data prior to encryption. Using the SNL DES ASIC on the front end as a randomizer introduces no significant delay to the host cryptosystem. In a similar vein, this device can be used as a random number generator.

One primary application of this device is as part of a cryptosystem based on the mode of operation known as *counter mode* or *filter generator*. In this type of cryptosystem (shown in Figure 8), a linear recurring sequence (LRS) generator produces a sequence, which is fed to a non-linear function. The purpose of the non-linear function (in this case, an SNL DES ASIC) is to mask the linearity properties of the LRS. The output of the DES ASIC is then combined with the data, through an Exclusive-OR operation.

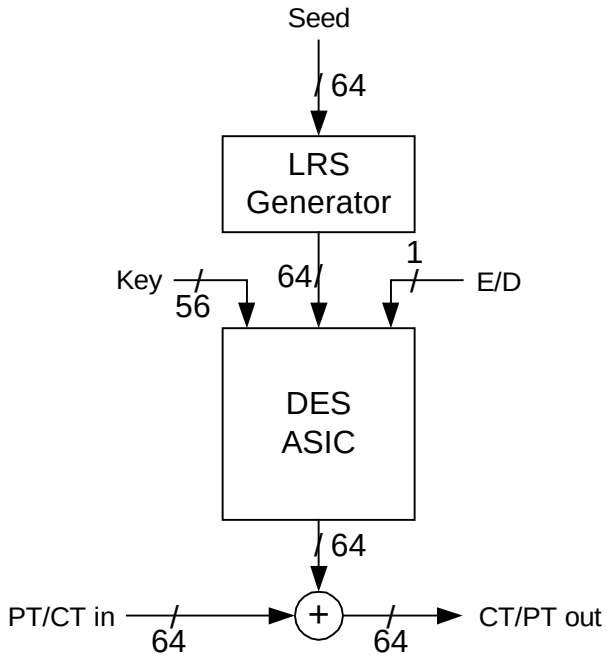


Figure 8. Encryption/Decryption for Counter Mode.

8.2 Triple DES

Triple DES employs the Data Encryption Standard algorithm in a way sometimes referred to as encrypt-decrypt-encrypt (EDE) mode. EDE mode using two keys, was proposed by W. Tuchman and summarized by Schneier in [11]. The incoming plaintext is encrypted with the first key, decrypted with the second key, and then encrypted again with the first key. On the other end, the received ciphertext is

decrypted with the first key, encrypted with the second key, and again decrypted with the first key to produce plaintext. If the two keys are set alike, it has the effect of single encryption with one key, thereby preserving backward compatibility. Two key, triple DES schemes (with 56 bit keys) can be cryptanalyzed using a *chosen plaintext* attack with about 2^{56} operations and 2^{56} words of memory [9]. Although in theory this is a weakness, Merkle and Hellman [9] state that in practice it is very difficult to mount a chosen plaintext attack against a DES cryptosystem. This makes two key, triple DES significantly stronger than two key, double DES, because an attack would now require 2^{112} operations (and no memory). (Two key, double DES is susceptible to a *known plaintext* attack with 2^{56} operations and 2^{56} words of memory [9].)

Triple DES can also be performed with three independent keys, using one key for each of the encryption and decryption operations. Triple DES with three independent keys gives a higher level of protection, requiring about 2^{112} operations and 2^{56} words of memory [11]. Again, with regards to compatibility, keys one and three could be set to the same value, to interoperate with Tuchman's two key, triple DES, or all three keys could be set alike to interoperate with single DES.

Because keys and control information march in lock step with the data, multiple SNL DES ASICs can be cascaded to provide the encrypt-decrypt-encrypt mode. Alternatively, data, key, and control information can be looped back, wrapping around the SNL DES ASIC to perform E-D-E triple DES using only one ASIC (see Section 8.2.2).

8.2.1 Cascading Multiple Devices

A string of three SNL DES ASICs can be laid out, connecting the output data, key out, and control information output pins of one ASIC to the input data, key in, and control information input pins on the next ASIC. To perform E-D-E triple DES, an inverter must be placed on the path of the encrypt/decrypt signal between ASICs. This way, the middle ASIC will always perform the opposite operation (encrypt or decrypt) from the first and last ASICs in the string. Programmable Logic Devices (PLDs) or SNL DES ASICs set to bypass mode can be used to provide the proper (18 clock tick) delay, so that the keys for the second and third encryption/decryption operations will arrive in synchronization with the appropriate data. An example of this is shown in Figure 9.

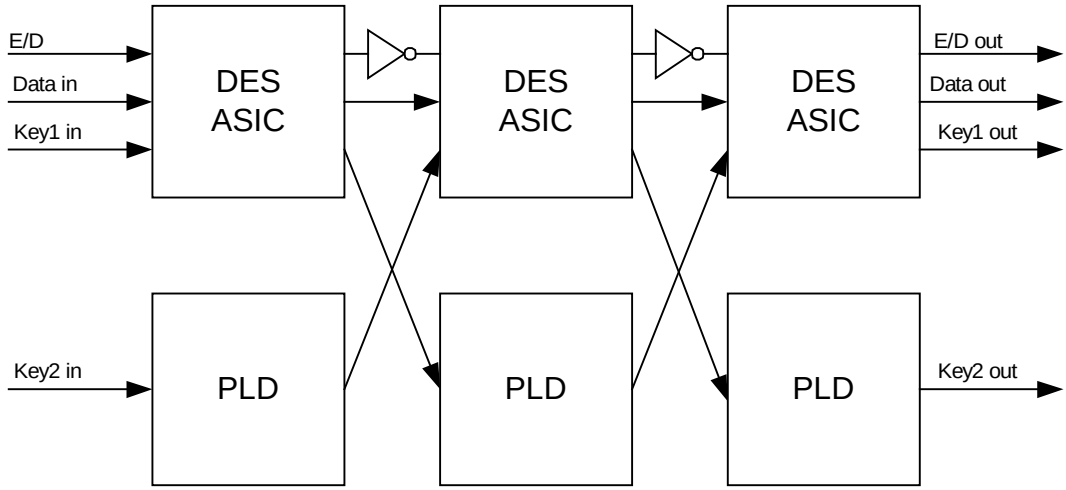


Figure 9. Cascaded, Multiple SNL DES ASICs.

8.2.2 Wrapping Around the Device

By applying appropriate glue logic, the SNL DES ASIC can be used to perform E-D-E triple DES by looping the data, key, and control information around the ASIC, processing the data three times. The glue logic will need to contain a two bit wide, 18 stage delay to count (in synchronization with the data, key, and control information) the number of times a given block of data has been processed. Logic will also be needed to invert the encrypt/decrypt bit between passes through the SNL DES ASIC. This method yields one third of the throughput of the cascaded method shown above. An example of the wrap-around implementation is shown in Figure 10.

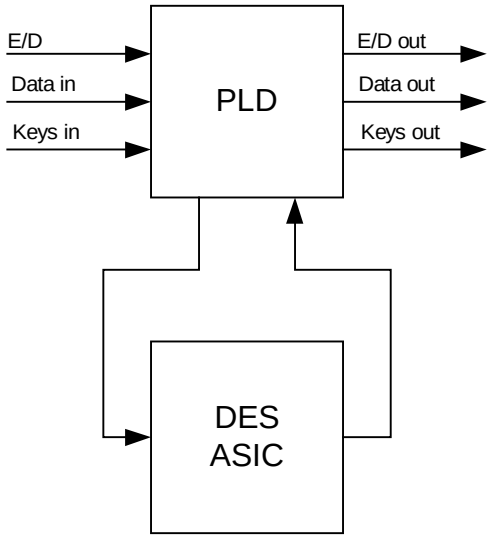


Figure 10. Wrap-Around Triple DES Implementation.

8.3 Parallel Operation

To achieve higher total throughput, multiple SNL DES ASICs can operate in parallel, with each ASIC processing a 64 bit block of the data stream. Figure 11 contains an example of cascaded devices, performing E-D-E triple DES on two blocks of data in parallel.

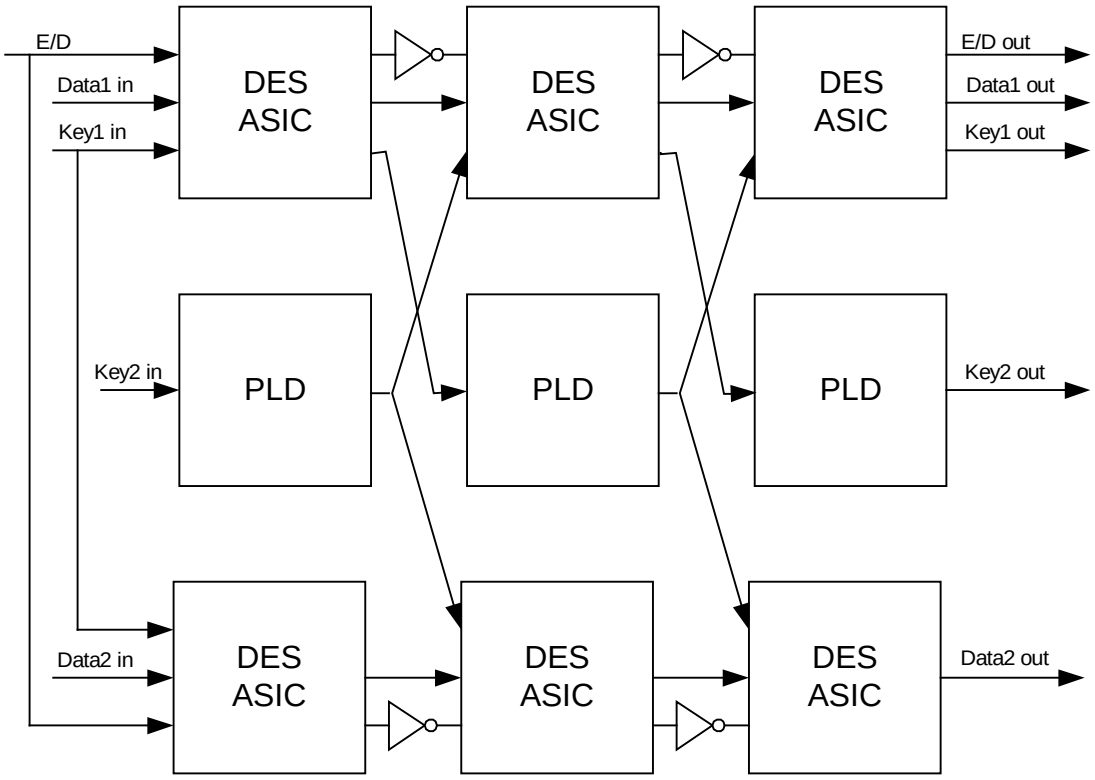


Figure 11. A Possible Implementation of E-D-E Mode Triple DES.

Because the SNL DES ASIC has tristate outputs, another mode of parallel operation can be used. This involves operating two SNL DES ASICs in parallel, but clocked on opposing clock edges. Data block one would enter the first DES ASIC on the rising edge of the first clock pulse, data block two would enter the second DES ASIC on the falling edge of the first clock pulse, data block three would enter the first DES ASIC on the rising edge of the second clock pulse, and so on. The output pins of each DES ASIC would feed a common data bus, 64 bits wide. In this way, a processed block leaves the set of DES ASICs on every clock transition.

Another use for this device is to keep encryption/decryption keys synchronized with the data in cascaded triple DES implementations (described above). At the highest data rates, programmable logic devices (PLDs) may not be able to keep pace with the SNL DES ASIC for passing keys with the data. Consequently, SNL DES ASICs

operating in bypass mode may be used to pass keys to subsequent encryption chips in step with the data.

8.4 Low Power Applications

Because the SNL DES ASIC is a fully static CMOS device, the power usage is proportional to operating frequency. Although at 105 MHz, the SNL DES ASIC consumes 6.5 Watts of power, the SNL DES ASIC can be operated at much lower clock rates, consuming very little power. This enables many low speed, extremely low power applications.

In the 1200 to 640,000 bits per second range, the DES ASIC consumes only microwatts of power, making it also well suited for low power applications running at low data rates. As tabulated elsewhere in this report (Table 2), the SNL DES ASIC operating at 10 KHz (640,000 bits per second, or ten 64 Kbps voice channels) only consumes 510 microwatts. Iterating around the ASIC to triple encrypt a single 64 Kbps voice channel, the SNL DES ASIC would need to operate at about 3 KHz, requiring well less than half of a milliwatt of power. Triple encrypting lower data rate channels (1200-28800 bps) requires even less power, enabling operation from a small battery or solar panel.

8.5 Encryption of SONET Synchronous Payload Envelopes (SPE)

Since the SNL DES ASIC has the capability to encrypt or bypass the processing of each input word, the SONET payload information is easily bypassed, allowing the Path, Section, and Line header overhead of each SPE to be processed in plaintext form. The encrypt, decrypt, and bypass inputs are used to control this function. Since neither the SONET Payload Envelope nor the SONET Transport Overhead is a multiple of 64 bits, some padding of these fields is necessary to pass through the DES ASIC pipeline. A more practical approach is to use a mode of operation that generates variable length "key stream" such as Counter Mode or Output Feedback Mode. In these modes the ciphertext or plaintext only passes through a mixer and does not pass through the fixed width DES pipeline. These design considerations are a function of the design of the DES block encryption algorithm, rather than this particular implementation in the SNL DES ASIC.

8.6 Encryption of ATM Cells

Again, the encrypt/decrypt or bypass capability of the SNL DES ASIC can be used to bypass the cell header and encrypt the cell payload. The Start of Cell (SOC) input is used to mark the beginning of each cell, and the Valid input can be used to mark idle cells that can be discarded at the output of the Encryption or Decryption pipeline. Since the ATM cell payload is an even multiple of 64 bit words (384 bits = 6 x 64 bit ciphertext or plaintext "words"), even ECB or CBC modes are easily implemented. The 32 bit cell header and the Header Error Checksum (HEC) must be extended/padded to a 64 bit length for passage through the pipeline in these

modes of operation. Of course, operation in Counter Mode or Output Feedback Mode, an exact number of keystream bits are "gated" into the mixer, eliminating the need for padding of header bits, since in this case the header passes through the mixer and not through the fixed width DES ASIC, as noted in the previous section.

8.7 Encryption of IP Datagrams

The encryption of variable length IP datagrams is subject to the same considerations as above. The datagram headers and payloads must either be padded to a multiple of 64 bit words, or a mode of operation must be employed that does not require the plaintext or ciphertext to pass through the fixed width pipeline. If padding is employed, the Start of Cell (SOC) and Valid inputs can be used to mark the beginning and end of the variable length packets. If a "keystream mode" is used, the bypass input can be used at input and output of the DES ASIC pipeline to throttle the amount of keystream generated and buffered, to match the throughput of the packet stream.

9.0 Lessons Learned

This is a brief summary of the lessons learned while designing and fabricating the SNL DES ASIC.

- Use design knowledge and engineering judgment to your advantage.
- Use structural VHDL for pad cells.
- Registers built structurally will facilitate clock trees, reset trees etc.
- Build your own clock tree -- Compass will corrupt your files for LVS.
- Pad cells have high capacitance inputs, drive them with high drives cells.
- Synthesize with realistic loads on internal blocks to prevent higher level surprises.
- For layout, partition into manageable cell areas.
- The Design Analyzer insert pads function does not work for input buffer pads, only clock and oscillator pads.
- In the Design Analyzer EDIF configuration, set "cell" not "net" for edifout_power_and_ground_representation.
- Pathfinder row numbers increase from bottom to top, not vice versa.
- Closing Compass applications after each major task will reduce run-time. (This raises the question of whether there might be a memory leak in Compass.)
- Create the top-level block netlist in a directory separate from subblock netlists.

10.0 Future Enhancements

Several enhancements to the SNL DES ASIC have been identified, since the initial fabrication runs. These enhancements, discussed in a paper presented at the 1999 CHES Workshop [13] and in a prior Sandia report [14], would increase throughput, aid in cascading devices, and improve flexibility and ease of use at the board level.

As built, the SNL DES ASIC has successfully demonstrated how a "heavyweight" encryption algorithm can be accelerated into the 10 Gbps arena. The identified enhancements should be considered for any future encryption ASIC, whether it be a follow-on DES or Advanced Encryption Standard (AES) ASIC.

Performance modifications (as also described in Section 6.3) include the chosen ASIC fabrication technology and geometry, length of the pipeline, the number of residing encryption engines, and I/O (input/output) buffers. Device yield could be increased through improved pin outs, thereby shrinking the silicon area. Finally, the next-level integration could benefit from modifications to the input and output signals and inclusion of certain key management operations into the ASIC.

Sandia National Laboratories MDL CMOS VI (0.6 um CMOS) was used to fabricate the SNL DES ASIC. This process was chosen for economy, locality, familiarity, and ease of use. Fabricating with this process imposed an upper bound on the ASIC throughput at 9.7 Gbps. For maximum throughput a newer, faster process would have reduced the pipeline delays. Any future design effort should compare the costs and benefits associated with Sandia's CMOS VII (0.35 um CMOS) against foundries processing smaller, faster features. Smaller feature sizes would offer additional opportunities to improve the design architecture. Architectural modifications would improve the performance with shorter pipeline delays and faster I/O buffers. The SNL DES ASIC was built with sixteen pipeline stages for the algorithm. Increasing the number of pipeline stages, or increasing the pipeline length, would distribute the combinatorial logic gates and reduce the minimum delay between registers. This would result in a greater data throughput at the cost of increasing the clock latency for any given datum. Fabrication with smaller features would not only permit higher clock rates, but also allow greater gate density and the possibility of parallel engines simultaneously processing distinct data. ASICs built with smaller geometries also operate at a lower voltage, requiring less power.

Further throughput enhancements would be realized by replacing the CMOS-level (0 to 5 Volt) I/O buffers with faster buffers such as low voltage differential signal (LVDS) interfaces. The CMOS buffers exhibit an upper frequency limit of 200 MHz, while LVDS work beyond 600 MHz. Also, the outputs of the SNL DES ASIC were tri-state drivers to allow the possibility of ping-ponging two devices for enhanced throughput. In retrospect it would be better to use simple output drivers that are faster than the tri-state outputs. Bringing out the clock, phased with the output

data would facilitate higher operational speeds and greater performance by enabling source synchronous clocking. (For a discussion of source synchronous clocking, see section 2.7.5.3.3 of the SAND2001-1062 report [14].)

Device yields on the SNL DES ASIC were reduced because of the large die area in silicon. This area could be reduced, by running two rings of I/O bondpads, rather than a single I/O ring. Reducing this area would produce more die per wafer and lower the losses-to-silicon-defect density. The alternative would be to fill the unused area in silicon with more functionality (such as key management functions and three instantiations of the encryption engine for Triple-DES operation within one chip), so the cost of low yield would be offset with increased functionality.

If three instantiations of the DES encryption engine were not integrated onto one chip, another useful design enhancement would be to bring out, as an output pin, the negation or inversion of the E/D (encrypt/decrypt) signal. This would make it easier to concatenate several DES ASICs to implement E-D-E mode of Triple-DES.

For encryption of Asynchronous Transfer Mode (ATM) communication sessions, six SNL DES ASICs could operate in parallel on 64 bit blocks to encrypt a 384 bit payload, achieving a data rate of 40 Gbs (OC-768). A new design using state-of-the-art commercial CMOS technology (.13 μm) should result in 40 Gbps operation, potentially making it possible to support OC-768 communication rates with only one DES ASIC in an encryptor. The authors expect a next generation CMOS process (under 90 nm) to produce an ASIC that may have a throughput approaching 160 Gbps. Again, multiple ASICs operating in parallel would support even higher data rates.

11.0 Conclusions

In this project, we explored how a representative "heavyweight," unclassified encryption algorithm could be optimized and pipelined for realization as a silicon ASIC. Initially simulated, the algorithm was implemented in reprogrammable logic and operation of the pipeline algorithm was tested. Later, a full ASIC was designed utilizing lessons learned from the earlier prototypes. The resulting device was proven to be the fastest implementation of its time, operating at encryption rates of over 6.7 Gbps. The result was not only the fastest known DES implementation, but one that could handle changes of the key on each clock cycle making it key agile. The SNL DES ASIC can support two- or three-key triple DES using a multiple cascaded ASIC configuration. It can also support very low power operation, as the design is completely synchronous. In the future, this chip design could support higher data rates. It has been estimated that using this design, over 40 Gbps can be supported in a .13 μm process technology. Finally, it is anticipated that this research can be applied to the recently approved AES algorithm (Rijndael) with similar results.

The Department of Energy (DOE) has granted Sandia permission to assert its copyright [12] on the VHDL for the SNL DES ASIC as SCR469.

12.0 References

1. Mark Bean, *Evaluation of Hardware Implementations of the DES Algorithm*, Informal Report, National Security Agency, Fort Meade, MD, 1996.

2. Data Encryption Standard (FIPS PUB 46-2), Federal Information Processing Standards Publication 46-2, National Bureau of Standards, Washington, D. C., December 30, 1993.

3. Joan Daemen and Vincent Rijmen, *The Design of Rijndael*, Springer-Verlag, Berlin, 2002.

4. DES Modes of Operation (FIPS PUB 81), Federal Information Processing Standards Publication 81, National Bureau of Standards, Washington, D. C., December 2, 1980.

5. Hans Eberle, A High-Speed DES Implementation for Network Applications, in *Advances in Cryptology – Crypto 92*, Vol. 740 of *Lecture Notes in Computer Science*, held in Santa Barbara, CA, August 16-20, 1992. Springer-Verlag, Berlin, 1993.

6. J. M. Green, and R. E. Mikawa, *RS096: Microcontroller Core Application Specific Integrated Circuit Design Process Development and Implementation Report*, SAND97-0267, Sandia National Laboratories, Albuquerque, NM, February 1997.

7. Guidelines for Implementing and Using the NBS Data Encryption Standard (FIPS PUB 74), Federal Information Processing Standards Publication 74, National Bureau of Standards, Washington, D. C., April 1, 1981.

8. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, FL, 1997.

9. Ralph C.Merkle, and Martin E. Hellman, On the Security of Multiple Encryption,in *Communications of the ACM*, Vol. 24, pp. 465-467, July 1981.

10. Perry J. Robertson, Robert L. Hutchinson, Lyndon G. Pierson, Thomas D. Tarman, and Edward L. Witzke, *Final Report and Documentation for the PLD11 Multipurpose Programmable Logic VME Board Design*, SAND99-0914, Sandia National Laboratories, Albuquerque, NM, April 1999

11. Bruce Schneier, *Applied Cryptography*, 2nd ed., John Wiley & Sons, New York, 1996.

12. D. Craig Wilcox, Data Encryption Standard (DES) ASIC VHDL Ver. 1.0, SCR469, August 12, 1999.

13. D. Craig Wilcox, Lyndon G. Pierson, Perry J. Robertson, Edward L. Witzke, and Karl Gass, A DES ASIC Suitable for Network Encryption at 10 Gpbs and Beyond, in *Cryptographic Hardware and Embedded Systems*, Vol. 1717 of *Lecture Notes in Computer Science*, held in Worcester, MA, August 12-13, 1999. Springer-Verlag, Berlin, 1999.

14. Edward L Witzke, Lyndon G. Pierson, Thomas D. Tarman, L. Byron Dean, Perry J. Robertson, and Philip L. Campbell, *Final Report for the 10 to 100 Gigabit/Second Networking Laboratory Directed Research and Development Project*, SAND2001-1062, Sandia National Laboratories, Albuquerque, NM, April 2001.

Appendix I: DES ASIC Data Sheets

In this appendix are the data sheets for the SNL DES ASIC and its employment in an instantiation of Triple-DES.

Sandia National Laboratories

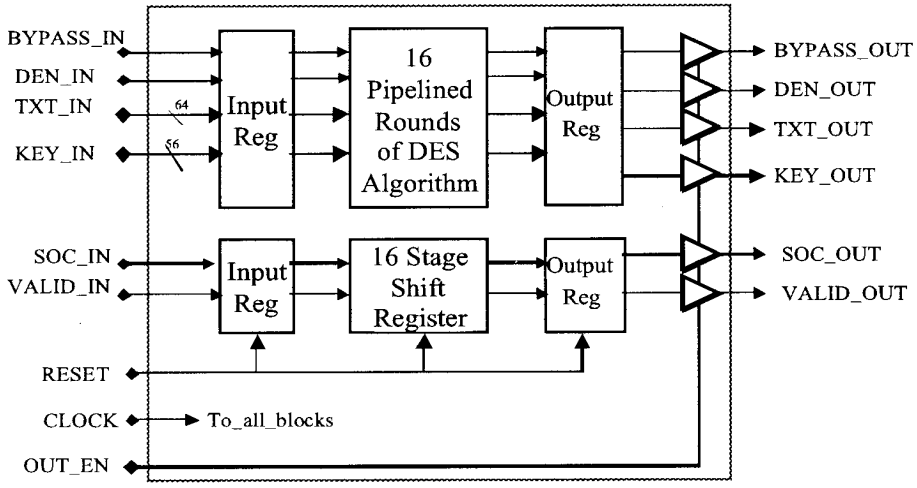
DES ASIC DATA SHEET

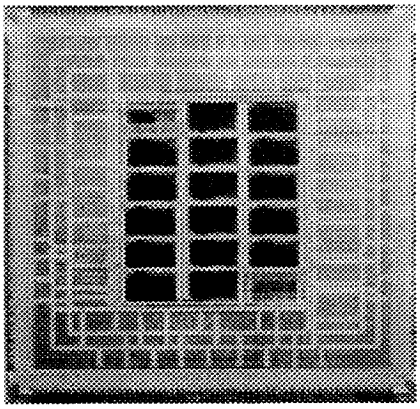
SEPT. 4, 1998

OVERVIEW:

The Data Encryption Standard (DES) as defined in the Federal Information Processing Standards (FIPS) Publication 46-1 is used for protecting data by cryptographic means. Sandia National Labs has implemented the DES algorithm in an Application Specific Integrated Circuit (ASIC). The design allows encryption, decryption, unique key input, or algorithm bypassing on each clock cycle. This is a 60K gate pipe-lined implementation having 251 IO signals and 68 power and ground pins. The chip has been fabricated in a 0.6 um CMOS process using a fully static design. This chip has been shown to operate as high as 105 MHz, yielding a single device throughput of 6.7 Gb/s. Six devices operating on an entire ATM Cell (384 payload bits in parallel) will yield OC-768c throughput of 40 Gb/s. While power dissipation at high throughputs is a challenge, the same device clocked at low speed can achieve encryption of multiple voice channels while consuming less than half a milliwatt of power.

FUNCTIONAL BLOCK DIAGRAM:





- Fully pipelined DES
- Triple-DES by cascading 3 devices
- 0.6 micron CMOS
- Die size: 11.1 mm square
- 319 total pins, 251 I/O
- 6.5 Watts @ 105 Mhz (6.7 Gb/s)
- Each clock cycle can:
  - Bypass
  - Encrypt/Decrypt
  - Establish new key
  - Flag Start of Cell (SOC)

OVERVIEW:

This module employs the Data Encryption Standard (DES) [1] as the encryption algorithm. The module uses it in a way sometimes referred to as encrypt-decrypt-encrypt (EDE) mode. EDE mode using two keys, proposed by W. Tuchman and summarized by Schneier*. The incoming plain text is encrypted with the first key, decrypted with the second key, and then encrypted again with the first key, achieving the increased security equivalent to 112 bit key. On the other end, the received cipher text is decrypted with the first key, encrypted with the second key, and again decrypted with the first key to produce plain text. If the two keys are set alike, it has the effect of single encryption with one key, thereby preserving backward compatibility.

FUNCTIONAL BLOCK DIAGRAM:

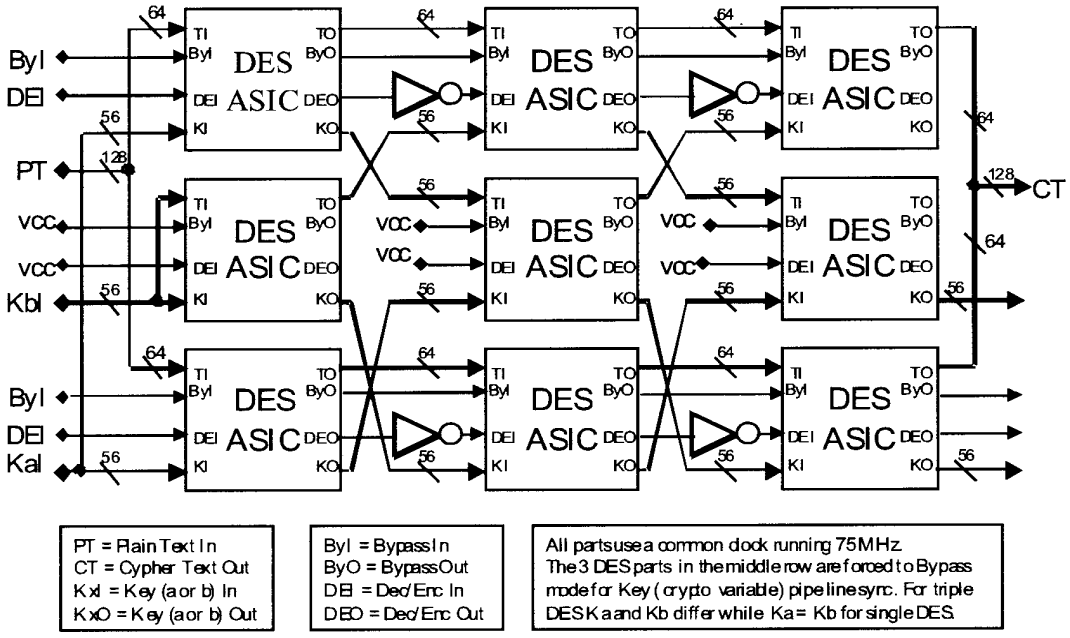


Figure 1. 10 Gb/s Triple DES module layout (embodiment 1).

*W. Tuchman, "Hellman Presents No Shortcut Solutions to DES," *IEEE Spectrum*, v. 16, n. 7, July 1979, pp.40-41.
B. Schneier, Applied Cryptography, John Wiley & Sons,Inc. 1994, pp.165-169, 241.

DISTRIBUTION:

2 Xilinx Inc.
Attn: D. Craig Wilcox
7801 Jefferson NE
Albuquerque, NM 87109

1	MS 0139	M. O. Vahle, 9900	1	MS 0874	D. W. Palmer, 1751
1	MS 0513	J. P. VanDevender, 1000	5	MS 0874	P. J. Robertson, 1751
1	MS 0630	M. J. Murphy, 9600	1	MS 0874	K. L. Gass, 1751
1	MS 0785	R. L. Hutchinson, 6516	1	MS 1072	K. K. Ma, 1735
1	MS 0785	P. L. Campbell, 6516	1	MS 9003	P. W. Dean, 8960
1	MS 0801	A. L. Hale, 9300	1	MS 9011	B. V. Hess, 8941
1	MS 0801	W. F. Mason, 9320	1	MS 9915	H. Y. Chen, 8961
1	MS 0801	M. R. Sjulin, 9330			
1	MS 0806	P. C. R. Jones, 9322			
1	MS 0806	C. D. Brown, 9322			
1	MS 0806	J. P. Long, 9322			
1	MS 0806	L. Stans, 9336			
1	MS 0806	J. P. Brenkosh, 9336			
1	MS 0806	J. M. Eldridge, 9336			
1	MS 0806	A. Ganti, 9336			
1	MS 0806	S. A. Gossage, 9336			
1	MS 0806	T. C. Hu, 9336			
1	MS 0806	B. R. Kellogg, 9336			
1	MS 0806	L. G. Martinez, 9336			
1	MS 0806	M. M. Miller, 9336			
1	MS 0806	J. H. Naegle, 9336			
1	MS 0806	R. R, Olsberg, 9336			
10	MS 0806	L. G. Pierson, 9336			
1	MS 0806	T. J. Pratt, 9336			
1	MS 0806	J. A. Schutt, 9336			
1	MS 0806	J. D. Tang, 9336			
1	MS 0806	T. D. Tarman, 9336			
1	MS 0806	L. F. Tolendino, 9336			
1	MS 0806	J. S. Wertz, 9336			
1	MS 0806	D. J. Wiener, 9336			
5	MS 0806	E. L. Witzke, 9336			
1	MS 0812	J. H. Maestas, 9334			
1	MS 0813	M. J. Ernest, 9333			

1	MS 9018	Central Technical Files, 8945-1
2	MS 0899	Technical Library, 9616