AES 128 Encryption/Decryption

David Leifker & Gentre Graham
Bradley University
Department of Electrical Engineering
Advisor: Dr. Vinod Prasad

Abstract

Advanced Encryption Standard (AES), a Federal Information Processing Standard (FIPS), is an approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a block cipher that can encrypt and decrypt digital information. The AES algorithm is capable of using cryptographic keys of 128, 192, and 256 bits, this project implements the 128 bit standard on a Field-Programmable Gate Array (FPGA) using the VHDL, a hardware description language. In June 2003, the National Security Agency (NSA) announced that AES-128 may be used for classified information at the SECRET level and AES-192/256 for TOP SECRET level documents.

Introduction

AES is an algorithm for performing encryption (and the reverse, decryption) which is a series of well-defined steps that can be followed as a procedure. The original information is known as plaintext, and the encrypted form as cipher text. The cipher text message contains all the information of the plaintext message, but is not in a format readable by a human or computer without the proper mechanism to decrypt it; it should resemble random gibberish to those not intended to read it. The encrypting procedure is varied depending on the key which changes the detailed operation of the algorithm. Without the key, the cipher cannot be used to encrypt or decrypt. In the past, cryptography helped ensure secrecy in important communications, such as those of government covert operations, military leaders, and diplomats. Cryptography has come to be in widespread use by many civilians who do not have extraordinary needs for secrecy, although typically it is transparently built into the infrastructure for computing and telecommunications (Wikipedia).

Objective

The design goal of this project was to create a demonstration of the AES for the end user and not for integration into a communication or data storage device; however this design could be modified to such ends. Since the goal was to create a demonstration, two human interface devices (HIDs), a regular PS2 keyboard and a 4x20 line LCD comprise the inputs and outputs to the system. A top level block diagram is shown in Figure 1 to illustrate.

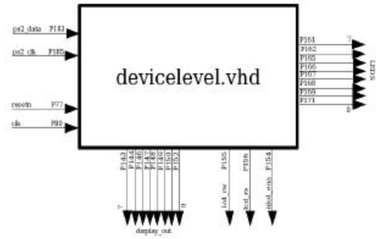

Figure 1: Device Level Block Diagram

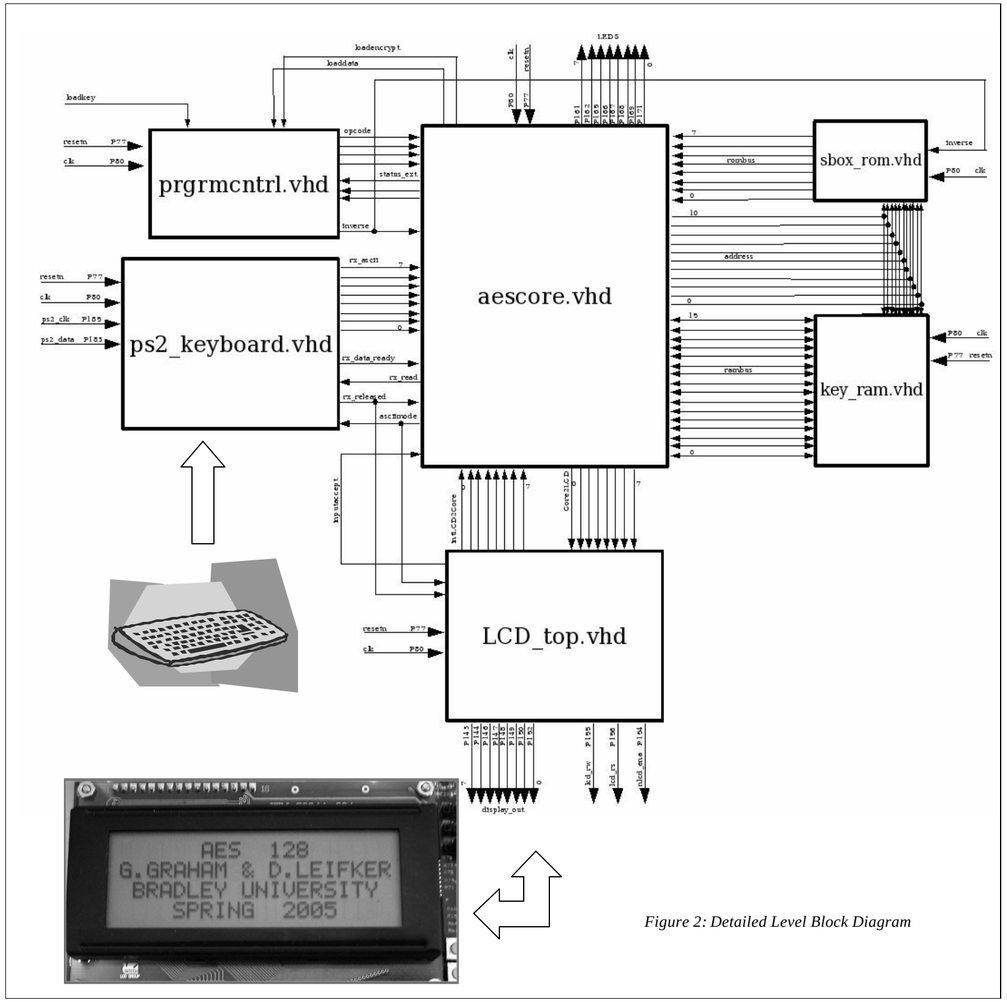

Figure 2: Detailed Level Block Diagram

Implementation

There are 3 main routines implemented to realize the AES algorithm: encryption, decryption, and key expansion routines. The VHDL code is organized into several modules shown in Figure 2. The overall design includes microprogramming concepts and is based on Harvard architecture. Harvard architecture includes separate program and data memory. The memory devices prgrmcntrl.vhd, sbox_rom.vhd, and key_ram.vhd utilize synchronous data transfer. In contrast, the ps2_keyboard.vhd and LCD_top.vhd modules are based on asynchronous data transfer due to the relative speed of the core devices to the lower speed of the HIDs. The sbox_rom.vhd module stores the SBOX and inverse SBOX tables which contain 512 bytes of data, for the substitution box operation referred to in the FIPS197 standard. The key_ram.vhd stores the key expansion, 176 bytes of data, which is calculated from the original 128bit key. This key allows the data to be converted from cipher text to plaintext and visa versa. Prgramcntrl.vhd contains the micro coded instructions of the 3 routines and thus controls the sequential operations of the core. The ps2_keyboard.vhd module decodes the scan codes generated by the keyboard into either ASCII or hexadecimal format for input into the core. LCD_top.vhd displays the user prompts as well as the input and output of the core.

Results

Encryption and decryption routines are fully functional at 50 and 100 MHz. The final product has a hard coded key because of the area constraints on the particular Xilinx Spartan3 FGPA device used in the project. The software generated key expansion was simulated and run on hardware without the keyboard input and LCD output. The test vector provided by FIPS 197 is displayed in Figure 3.

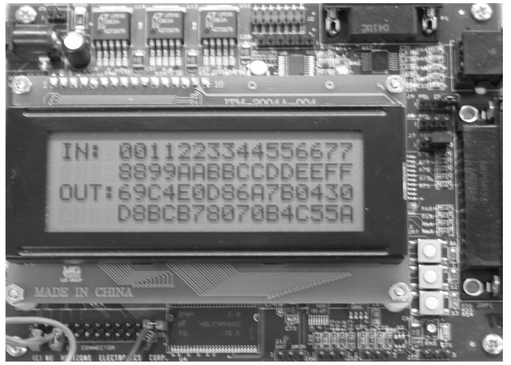

Figure 3: Development Board Displaying Standard Test Vector

Reference

Federal Information Processing Standard (FIPS) 197
http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf