Advanced Encryption Standard / Rijndael IP Core

Author: Rudolf Usselmann
rudi@asics.ws
www.asics.ws

Rev. 1.0
November 9, 2002

Revision History

Rev.	Date	Author	Description
1.0	11/9/02	Rudolf Usselmann	Initial Release

1

Introduction

Simple AES/Rijndael IP Core. I have tried to create a implementation of this standard that would fit in to a low cost FPGA, like the Spartan IIe series from Xilinx, and still would provide reasonably fast performance.

This implementation is with a 128 bit key expansion module only. Implementations with different key sizes (192 & 256 bits) and performance parameters (such a a fully pipelined ultra-high -speed version) are commercially available from ASICS.ws (www.asics.ws).

This document will describe the interface to the IP core. It will not talk about the AES standard itself.

(This page intentionally left blank)

2

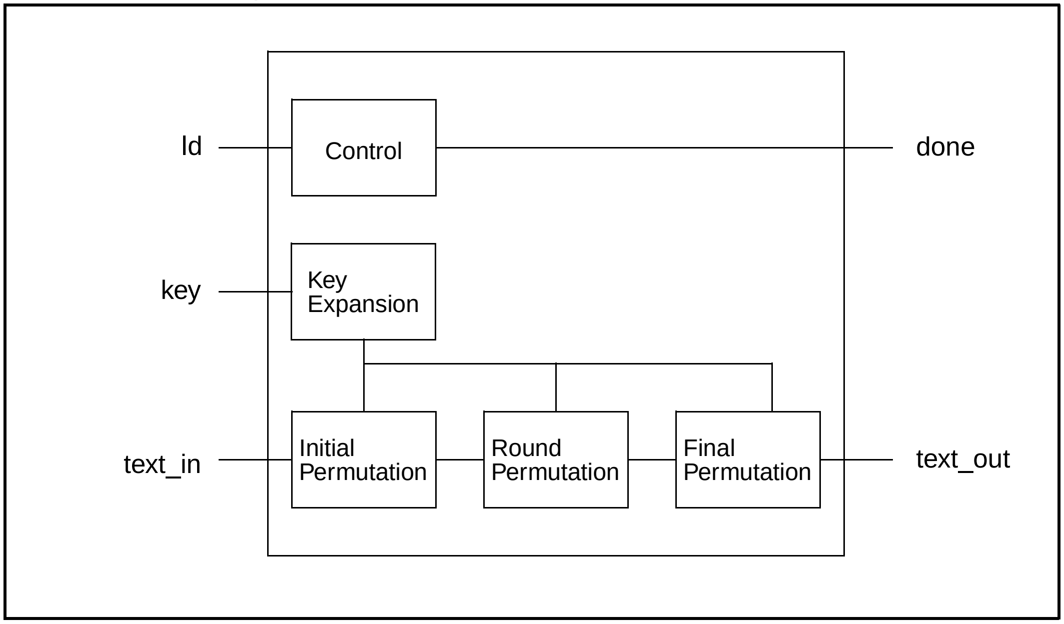
Architecture

The AES Rijndael core consists of two blocks: 1) The AES Cipher block with performs decryption; 2) The AES Inverse Cipher block with performs encryption. Both blocks instantiate the same key expansion block.

2.1. AES Cipher Core

Below figure illustrates the overall architecture of the AES Cipher core.

Figure 1: Cipher Core Architecture Overview

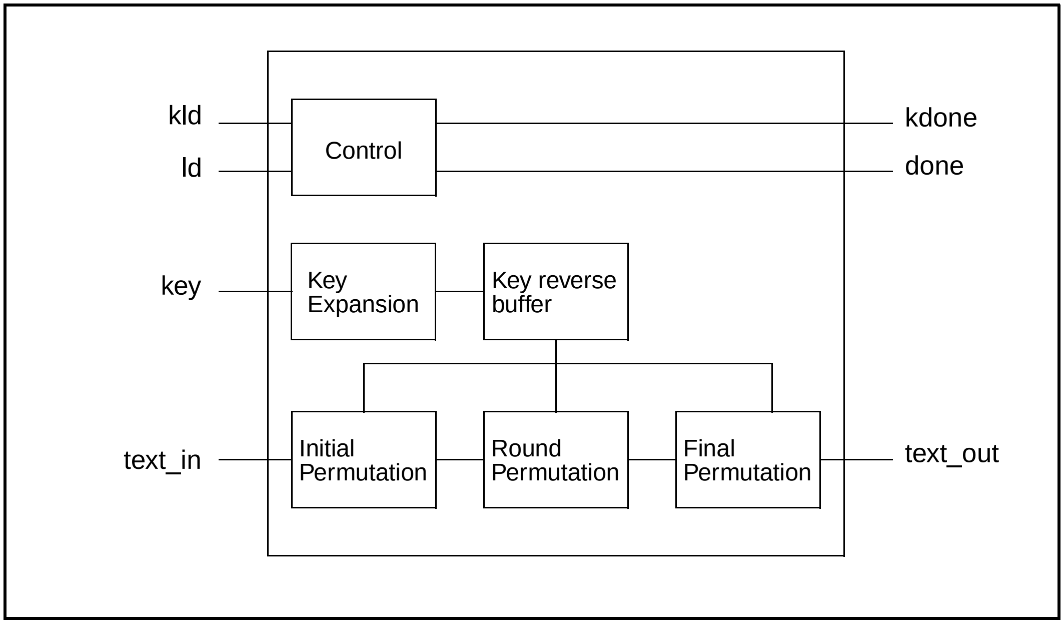


The AES cipher core consists of a key expansion module, an initial permutation module, a round permutation module and a final permutation module. The round permutation module will loop internally to perform 10 iteration (for 128 bit keys).

2.2. AES Inverse Cipher Core

Below figure illustrates the overall architecture of the AES Inverse Cipher core.

Figure 2: Inverse Cipher Core Architecture Overview



The AES inverse cipher core consists of a key expansion module, a key reversal buffer, an initial permutation module, a round permutation module and a final permutation module.

The key reversal buffer first stores keys for all rounds and the presents them in reverse order to the rounds.

The round permutation module will loop maternally to perform 10 iteration (for 128 bit keys).

3

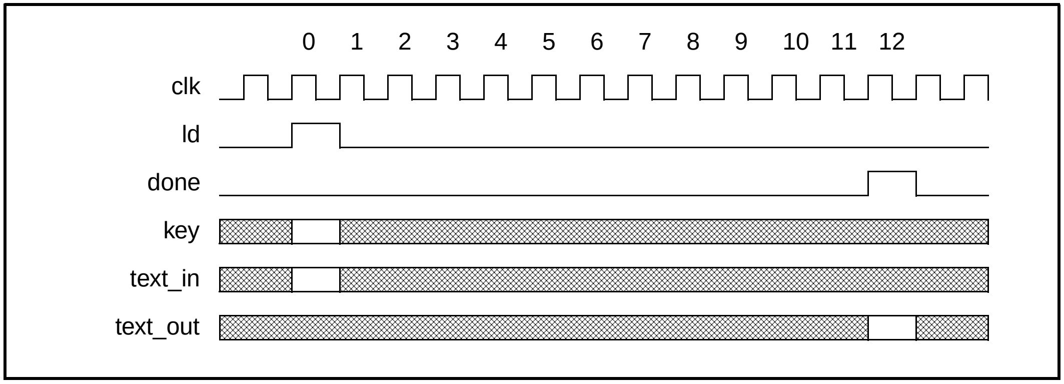
Operation

3.1. AES Cipher Core

The forward cipher block can perform a complete decrypt sequence in 12 cycles (10 cycles for the 10 round, plus one cycle for initial key expansion, and one cycle for the output stage).

The forward cipher block accepts a key and the plain text at the beginning of each decrypt sequence. The beginning is always indicated by asserting the ‘ld’ pin high. When the core completes the decryption sequence it will assert the ‘done’ signal for one clock cycle to indicate the completion. The user might chose to ignore the ‘done’ output and time the completion of the decryption sequence externally.

Figure 3: AES cipher core interface timing



3.2. AES Inverse Cipher Core

The inverse cipher block can perform a complete encrypt sequence in 12 cycles (10 cycles for the 10 rounds, plus one cycle for initial key loading, and one cycle for the output stage).

The inverse cipher, however, requires that the key is loaded before encryption can be performed. This is because it uses the last expanded key first and the first expanded key last. Once the key has been loaded, the expanded versions are generated and stored in an internal buffer. The expanded keys can be reused for subsequent encryption sequences for the same key. The key is loaded when the ‘kld’ signal is asserted high. Once key expansion sequence is completed, the ‘kdone’ signal will be asserted for one clock cycle.

The beginning of the actual encrypt sequence is always indicated by asserting the ‘ld’ signal high. When the core completes the encryption sequence it will assert the ‘done’ signal for one clock cycle. The user might chose to ignore the ‘done’ output and time the completion of the encryption sequence externally.

The key loading and encryption sequences can not happen in parallel. A key must always be loaded before the encryption sequence can be performed.

Figure 4: AES inverse cipher key load timing

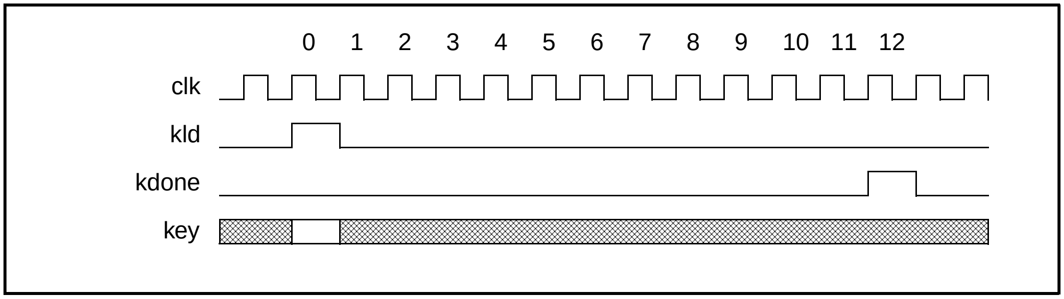
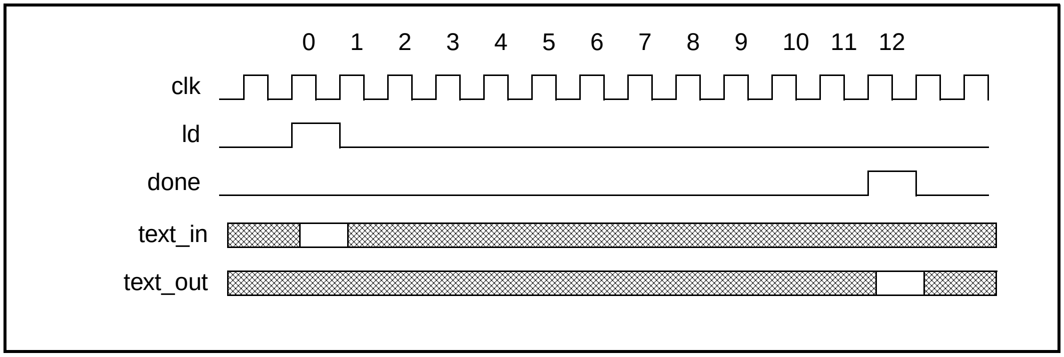


Figure 5: AES inverse cipher text block interface timing



4

Core IOs

4.1. Interface IOs

Table 1: Core Interfaces

Name	Width	Direction	Description
AES Cipher Core Interface			
clk	1	I	core clock
rst	1	I	active low synchronous reset
ld	1	I	load
done	1	O	done
key	128	I	key
text_in	128	I	input text block
text_out	128	O	output text block
AES Inverse Cipher Core Interface			
clk	1	I	core clock
rst	1	I	active low synchronous reset
kld	1	I	key load
kdone	1	O	key done
ld	1	I	text load
done	1	O	text done
key	128	I	key
text_in	128	I	input text block
text_out	128	O	output text block

(This page intentionally left blank)

Appendix A

File Structure

This section outlines the hierarchy structure of the AES Rijndael IP Core Verilog Source files.

Figure 6: AES Cipher Core Hierarchy Structure

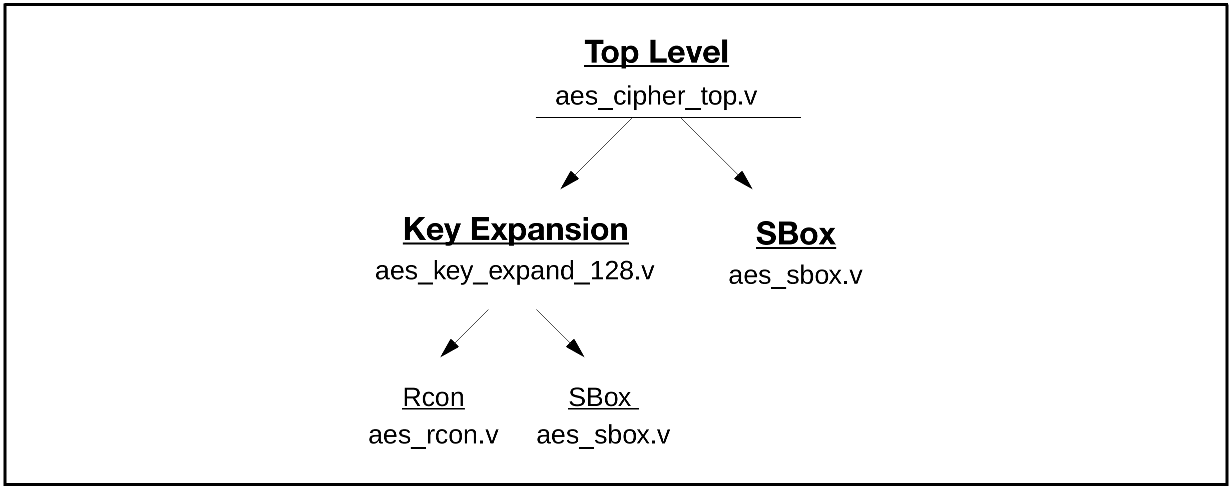
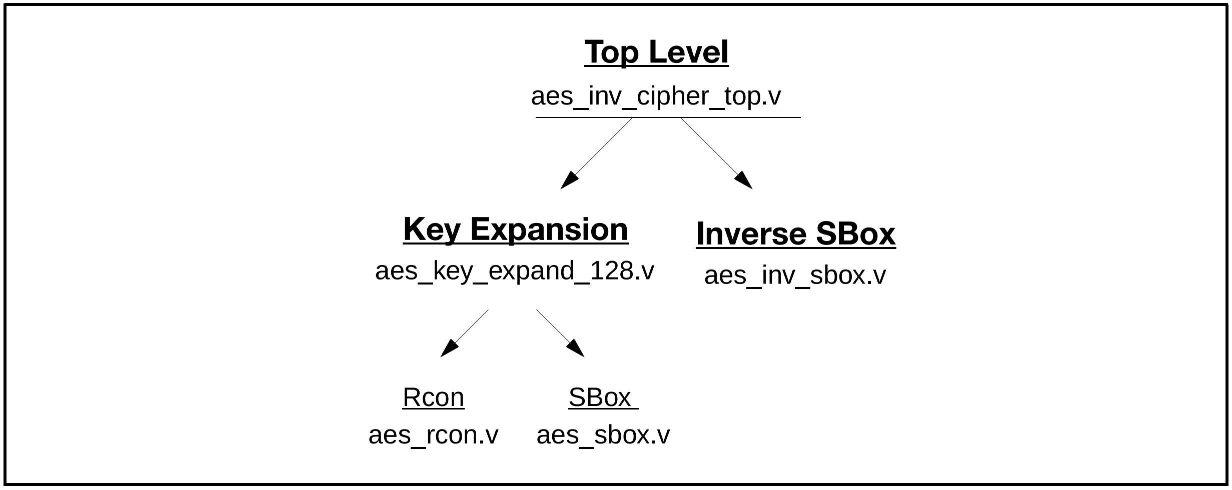


Figure 7: AES Inverse Cipher Core Hierarchy Structure


Top Level
aes_inv_cipher_top.v
Key Expansion
aes_key_expand_128.v
Inverse SBox
aes_inv_sbox.v
Rcon
aes_rcon.v
SBox
aes_sbox.v