aes-128 encryption chip

1.description

This chip achieves capability to encode and decode continuous data based on 128-bit AES encryption with user-configurable options and default state. The chip makes provision that set mode, secret key and output speed to satisfy variable situations and it is convenient to use a pair of chips to accomplish encoding and decoding between the sender and receiver. The code and framework is scalability for extensive instructions and verilog code will be open on github.

1.1 module schematic

The chip contains an input port, an output port, an instruction set and an 128-bit aes core.

1.2 chip pins introduction

vdd1: 3.3V power to support pad

vss1: gnd for vdd1

vdd2: 1.8V power to support chip core

vss2: gnd for vdd2

clk: external clock for the whole chip

rst: signal to reset the whole chip

cu: pin to select chip state(configurable mode or default mode)

id: in configurable mode, it indicates current input is instruction or data;

in default mode, it indicates current working state is encoding or decoding

input[7:0]: 8-bit input data port

shi: shake hand signal to indicate valid data

output[7:0]: 8-bit output data port

sho: shake hand signal to indicate valide data

1.3 parameter

chip area:

chip package:

chip clock:

chip power:

chip throughput:

1.4 128-bit aes

This chip selects and implements the 128-bit aes-128 global benchmark standard as its data encryption and decryption source. It is based on open source on github and we will give an introduction about this latter.

2.function description

In this chapter, users will learn how to use this chip to achieve 128-bit aes encryption. It will include transfer protocol and pin configuration.

2.1 user input port transfer protocol

The chip requests users to give a shake hand signal (in short as shi) to tell the chip the current data is valid. When users change input data, users should flip the shi signal along. If the shi signal is not changing, it means there is no new data in. It is shown as the following pictures.

2.2 user output port transfer protocol

The chip send dealt data through the output port module at the speed set in the chip. And the chip will give a square wave whose duty ratio is 50% to indicate current valid data. The signal will be high when data changes and will be low when at the middle of transfer period. The constantly low sho signal indicates there is no new data be exported. It is shown as the following pictures.

2.3 user configurable options

When pin cu is high, chip is in configurable mode. In this mode, users can set parameters to set encryption mode, secret key replacement and output speed.

When pin id is high configurable mode will accept instructions as following 2.3.1 to 2.3.3.

2.3.1 mode selection

In configurable mode with high id signal, users can use the instruction “MODx” to change the chip working state. Users should input characters “MOD” and parameter x in order in four transfer periods to config the chip. Parameter x can be character “E”, which means encoding state, or character “D”, which means decoding state.

In configurable mode, if users do not use this instruction, the default state is decoding working.

2.3.2 secret key replacement

In configurable mode with high id signal, users can use the instruction “KEYx” to change the chip secret key. Users should input characters “KEY” and parameter x in order in four transfer periods to config the secret key. Parameter x can be character “F”, “Q” or “N”.

Parameter “F” means users will input 128-bit secret key after the instruction “KEYF” transfer. Users can config own secret key to make aes-128 encryption using “KEYF” and 16 key inputting periods.

Parameter “Q” means users will input 32-bit key to change the default key in the chip to achieve a safer situation. Users can input “KEYQ” and then input 32-bit key in 4 transfer periods. The chip will make up the 128-bit secret key but users will not know the whole secret. So we suggest that users can use a pair of chips to finish encoding and decoding.

Parameter “N” means users just use the default secret key stored in the chip. Users can not know the exact secret key but it is convenient to config the chip and use it. Users just input characters “KEYN” and the configuration is finished.

In configurable mode, if users do not use this instruction, the key will be default key stored in the chip.

2.3.3 output speed

In configurable mode with high id signal, users can use the instruction “SPDx” to set the time periods taken by an output transfer. Users should input characters “SPD” and hex parameter x in four transfer periods to config the output speed.

Parameter x can be number 0x02 from number 0x0f. And it means the outport will take 2^x clk periods to transfer 1 byte data.

In configurable mode , if users do not use this instruction, the parameter will be 0x04 and it means output port uses 16 clk periods to send 1 byte data.

Users should pay attention that the outport speed can not be lower than input speed, for this chip has no storage function but can communicate continuously.

2.3.4 data transfer

When pin id is low in configurable mode, users can input prepared data for dealing one byte by one byte in order. Users should pay attention that thanks to aes-128 standard the smallest unit of data should be 128-bit. So users can input n\*128bits data continuously to the chip.

2.4 default(usual) mode

When pin cu is low, chip is in default mode.This mode is made for users to use the chip directly without configuration.

Pin id will select the working state. If pin id is high, the chip works for encoding. And if pin is low, the chip works for decoding. The chip will use the default secret key stored in the chip to make aes-128 encryption. The outport will take 16 clk periods to send 1 byte data.

2.5 mode change

In configurable mode, if users want to use default mode, users should reset pin rst which is low-active and put the pin cu low. The chip can work in default mode.

In default mode, if users want to use configurable mode, users should stop data transfer and put cu high. The chip can work in configurable mode.

3.example for users