Table of Contents

AES Blocks

1.1 Add_round_key:

1.1.1 Module Description:

add round key is one of the round operations of the AES algorithm it performs xor operation between input state matrix (text bytes) and cipher key which is generated and expanded each round using key expansion algorithm described in the fips standard and will be illustrated in this design document.

Note: In the port mapping table instead of writing all the bytes in the block diagram, only multiple bytes were included to decrease the table size. The module

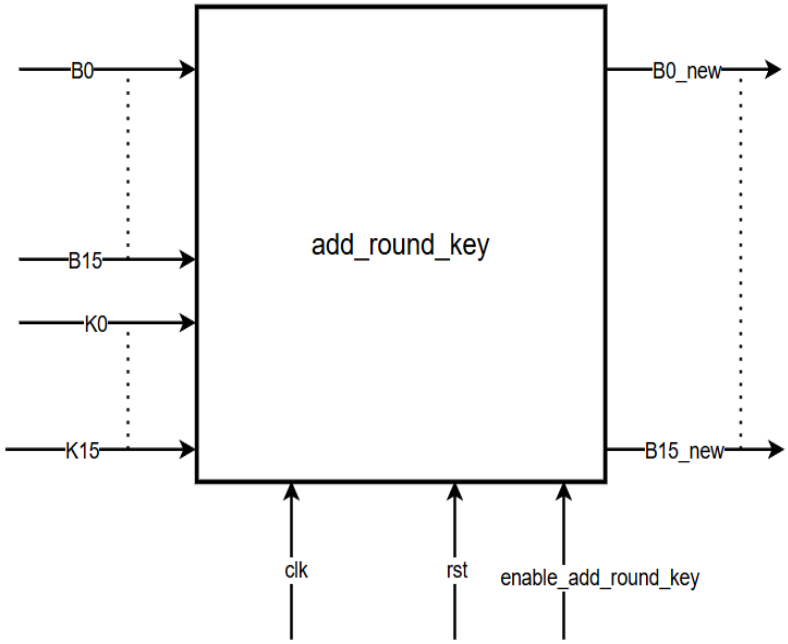

Figure 0-1 add_round_key block diagram

1.1.2 Port Mapping

Port Name	Port Width	Port Direction	Description
B0	8	Input	Input text byte 0
B1	8	Input	Input text byte 1
B15	8	Input	Input text byte 15
K0	8	Input	Input key byte 0
K1	8	Input	Input key byte 1
K15	8	Input	Input key byte 15
clk	1	Input	Input clk signal
rst	1	Input	Input asynchronous reset (negative edge)
enable_add_round_key	1	Input	Input enable from top level controller
B0_new	8	Output	Output byte 0
B1_new	8	Output	Output byte 1
B2_new	8	Output	Output byte 2
B3_new	8	Output	Output byte 3
B4_new	8	Output	Output byte 4

1.2 Counter

1.2.1 Module Description:

Counter module used to count the number of rounds , the number of rounds in AES algorithm depends on the key size , in this design there are 1 initial round, 9 main rounds and 1 final round.

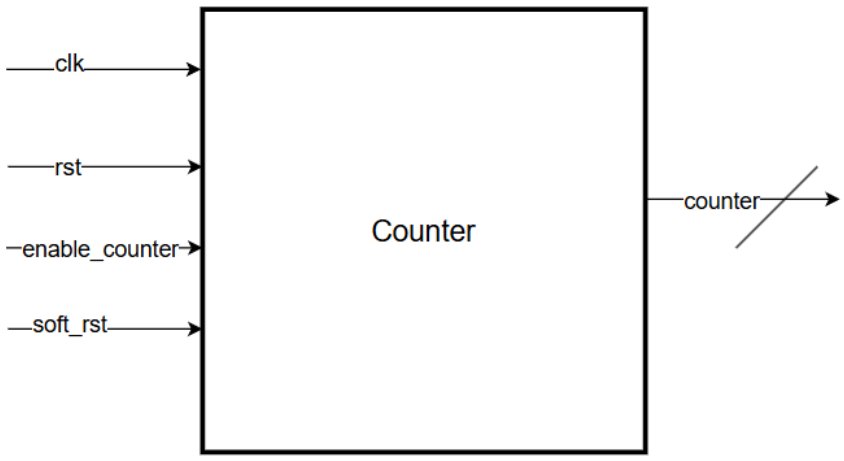

Figure 0-2 Counter block diagram

1.2.2 Port Mapping:

Port Name	Port Width	Port Direction	Signal description
clk	1	Input	clk signal
rst	1	Input	Asynchronous reset (active low)
enable_counter	1	Input	Counter enable signal
soft_rst	1	Input	Soft reset to reset the counter during the operation of the IP
Counter	4	Output	Output count

1.3 Key_expansion

1.3.1 Module Description:

The key expansion modules generates new key each round based on the previous round keys and the current round number. The round number controls a function called G function which generates a constant Rconn used in the calculations of the new key.

The first column of the key matrix (K0,K1,K2,K3) are calculated using the final row of the key matrix of the previous round by using one cyclic shift on (K12,K13,K14,K15) then performing byte substitution and xoring with the previous first column.

The remaining 3 columns of the key matrix are calculated by xoring the corresponding column from the previous key matrix with the previous column of the new key matrix.

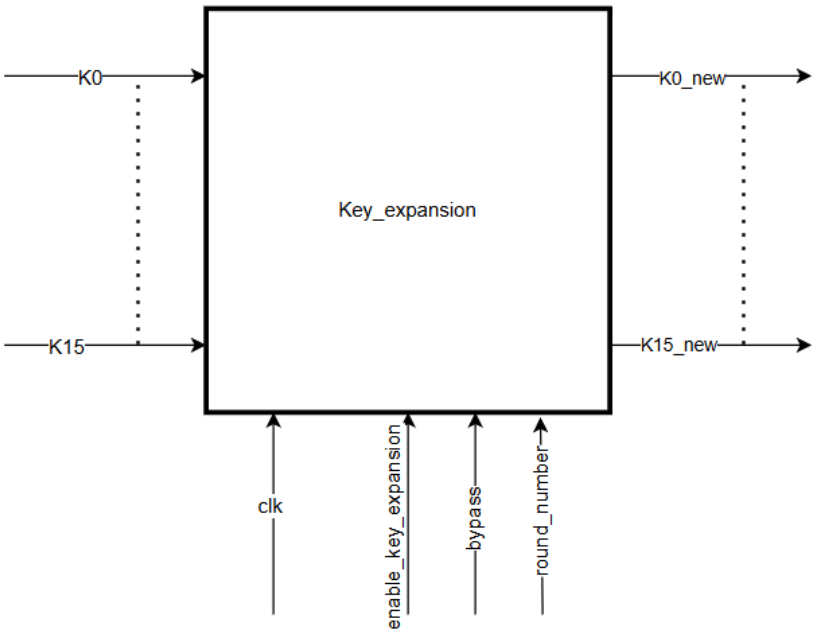

Figure 0-3 Key_expansion block diagram

1.3.2 Port Mapping:

Port Name	Port Width	Port Direction	Signal Description
K0	8	Input	Input key byte (same for K1-K15)
clk	1	Input	Input Clock
rst	1	Input	Asynchronous reset (active low)
bypass	1	Input	Bypass signal to make the module operate as buffer in the rounds where key expansion is not needed (Initial Round)
enable_key_expansion	1	Input	Enables the operation of the block
round_number	4	Input	Counter output which is used to count the round number
K0_new	8	Output	Output Key byte (same for K1-15)

1.4 Mix Columns

1.4.1 Module Description:

In the mix column modules operations are performed on the columns of the state matrix by multiplying a pre defined matrix from fips standard with the state matrix to generate the new corresponding bytes the multiplication is performed over GF(2) where each addition is an xor operation.

There are 3 numbers in the pre defined matrix 1,2,3 each multiplication is performed over GF(2). If the byte is multiplied by 1 the resulting byte is the same, However in case of multiplication by 2 it depends on the Msb if it is equal to 1 the byte is shifted left then xor operation is performed with a constant from the standard. In case of multiplication by 3 multiplication by 2 is performed then xor operation is calculated with the original byte.

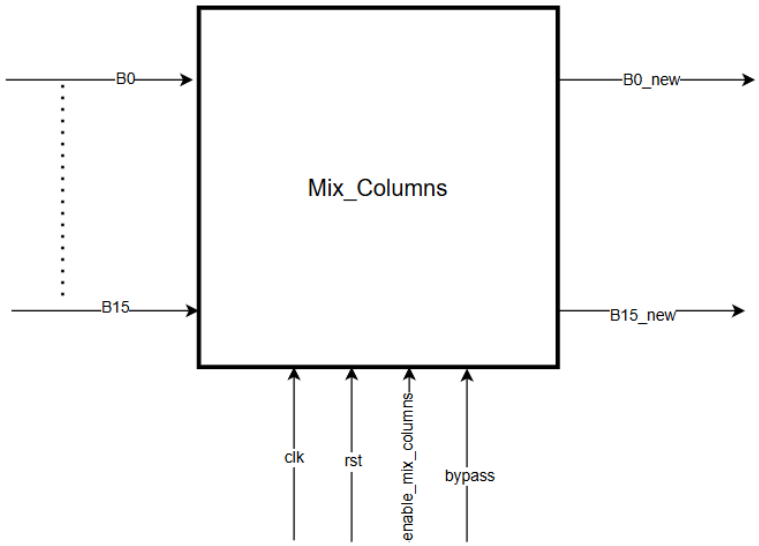

Figure 0-4 Mix columns block diagram

1.4.1.1 Port Mapping:

Port Name	Port Width	Port Direction	Signal Description
B0	8	Input	Input state byte (same for B1-B15)
clk	1	Input	Input clock
rst	1	Input	Asynchronous active low reset
enable_mix_columns	1	Input	Enable of mix columns from the top level controller
bypass	1	Input	Bypass to make the module act as buffer
B0_new	8	Output	Output state byte(same for B1-B15)

1.5 Shift Rows

1.5.1 Module Description:

Shift Rows module performs the shift rows operation in the AES round operations, the first row of the state matrix is kept the same, the second row is shifted one cyclic shift. Furthermore, the third row is shifted 2 cyclic shifts. Lastly, the fourth row is shifted 3 cyclic shifts.

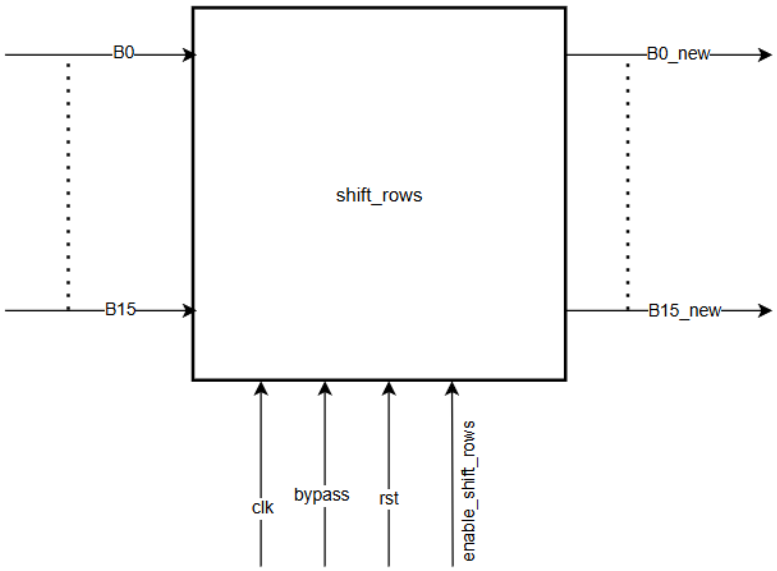

Figure 0-5 Shift rows block diagram

1.5.2 Port Mapping

Port Name	Port Width	Port Direction	Signal Description
clk	1	Input	System clk
rst	1	Input	Asynchronous active low rst
bypass	1	Input	Bypass to make the module act as buffer
enable_shift_rows	1	Input	enable shift rows from top level controller

B0	8	Input	Input text byte same for (B1-B15)
B0_new	8	Output	Output text byte after shifting rows same for (B1_new-B15_new)

1.6 S_Box

1.6.1 Module Description:

S_box module is the implementation of the s_box in the fips standard which is used in byte substitution, the s_box will be used in sub_byte module to substitute bytes to perform byte substitution on the state matrix. Sbox is built as a LUT.

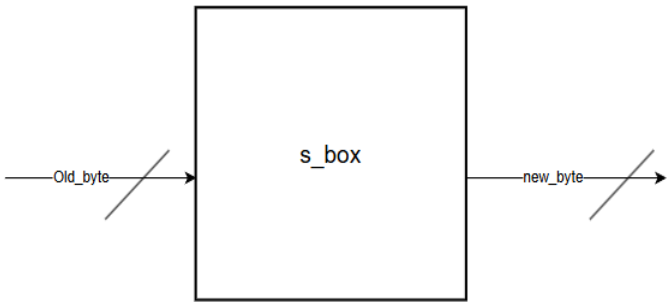

Figure 0-6 s_box block diagram

1.6.2 Port Mapping:

Port Name	Port Width	Port Direction	Signal Description
old_byte	8	Input	Byte before substitution from sbox
new_byte	8	Output	new byte after sub

1.7 Sub Byte:

1.7.1 Module Description:

The Sub Byte module instantiates S_Box module to substitute the 16 bytes of the state matrix according to the standard, this is one of the round operations of the AES Algorithm.

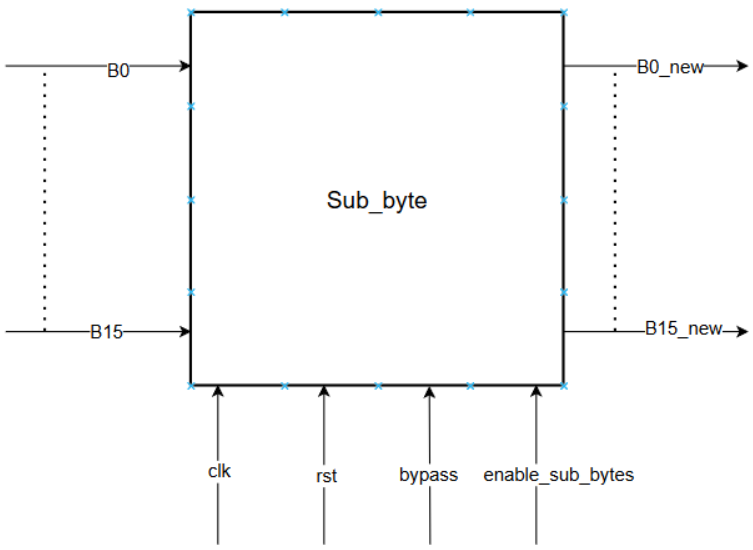

Figure 0-7 Sub byte block diagram

1.7.2 Port Mapping:

Port Name	Port Width	Port Direction	Signal Description
clk	1	Input	System clk
rst	1	Input	Asynchronous active low rst
bypass	1	Input	Bypass to make the module act as buffer

enable_sub_bytes	1	Input	enable sub_bytes from top level controller
B0	8	Input	Input text byte same for (B1-B15)
B0_new	8	Output	Output text byte after shifting rows same for (B1_new-B15_new)

1.8 Top level controller:

1.8.1 Module Description:

The top-level controller implements the FSM, that controls the operations of the AES algorithm. It provides all control signals for the round operations of the AES such as sub byte, mix columns, shift rows, add round key. In addition to indicating if the AES is operating in the initial round or the main rounds or the final rounds. The controller is also responsible for the soft reset signal which resets all the internal signals of the design to make the design capable of encrypting back-to-back blocks of text based on the input valid signal.

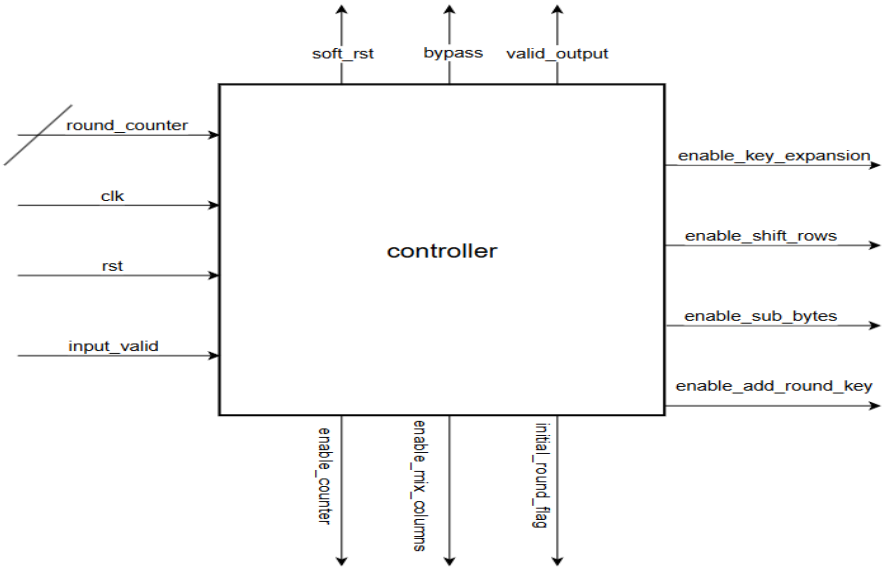

Figure 0-8 Controller block diagram

1.8.2 Port Mapping:

Port Name	Port Width	Port Direction	Signal Description
clk	1	Input	System Clock
rst	1	Input	Asynchronous Reset Active Low
input_valid	1	Input	Input port to the top-level design indicating the input 128-bit text is valid
round_counter	4	Input	Counter to count the number of rounds of the AES
enable_key_expansion	1	Output	enable key expansion module
enable_shift_rows	1	Output	enable shift_rows module
enable_sub_bytes	1	Output	enable sub_bytes module
enable_add_round_key	1	Output	enable add_round_key module
enable_counter	1	Output	enable_counter module which counts the number of rounds
enable_mix_columns	1	Output	enable mix_columns_module
initial_round_flag	1	Output	used as mux select in top level design
soft_rst	1	Output	used to reset the round counter after finishing the encryption process
valid_output	1	Output	used to indicate the completion of the encryption process
bypass	1	Output	used to make the design modules operate as buffers

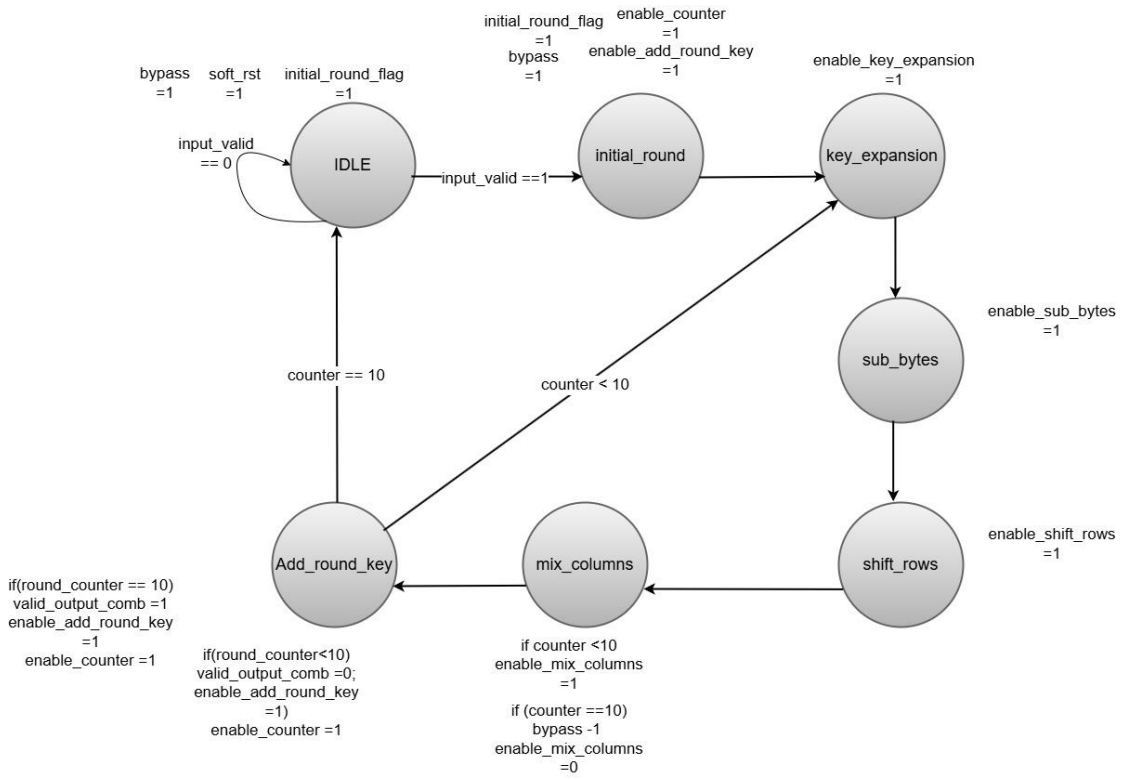

Figure 0-9 Controller FSM

1.9 Top Level Module:

1.9.1 Module Description:

Top Level module is the integration of all the sub modules of the design with additional blocks to ensure the proper operation of the AES such as some multiplexing and wiring, the wiring doesn't involve any additional hardware. The Design requires 52 clock cycles from the input valid signal assertion till the output valid signal assertion. Since the main rounds takes 5 clock cycles, 1 cycle for each round operation. Therefore, 5x10 = 50 clock cycles. In addition to, the initial round and the transition from IDLE to initial round which results in 52 cycles needed for the completion of the encryption process.

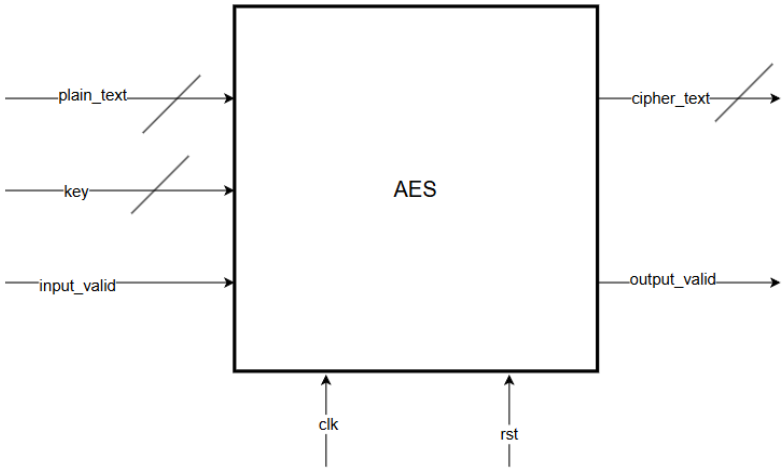

Figure 0-10 AES Top Module

1.9.2 Port Mapping:

Port Name	Port Width	Port Direction	Signal Description
plain_text	128	Input	input block text required to be encrypted
key	128	Input	initial key which is input to the AES block
clk	1	Input	System clock
rst	1	Input	Asynchronous reset (Active Low)
input_valid	1	Input	input indicating the plain_text is valid for AES operations
cipher_text	128	Output	Output text after the completion of the encryption process
output_valid	1	Output	valid signal indicating that the cipher_text signal is valid and AES operations is completed