AES-128 Hardware Implementation

➢ Main system characteristics

There are 2 options for the input format:

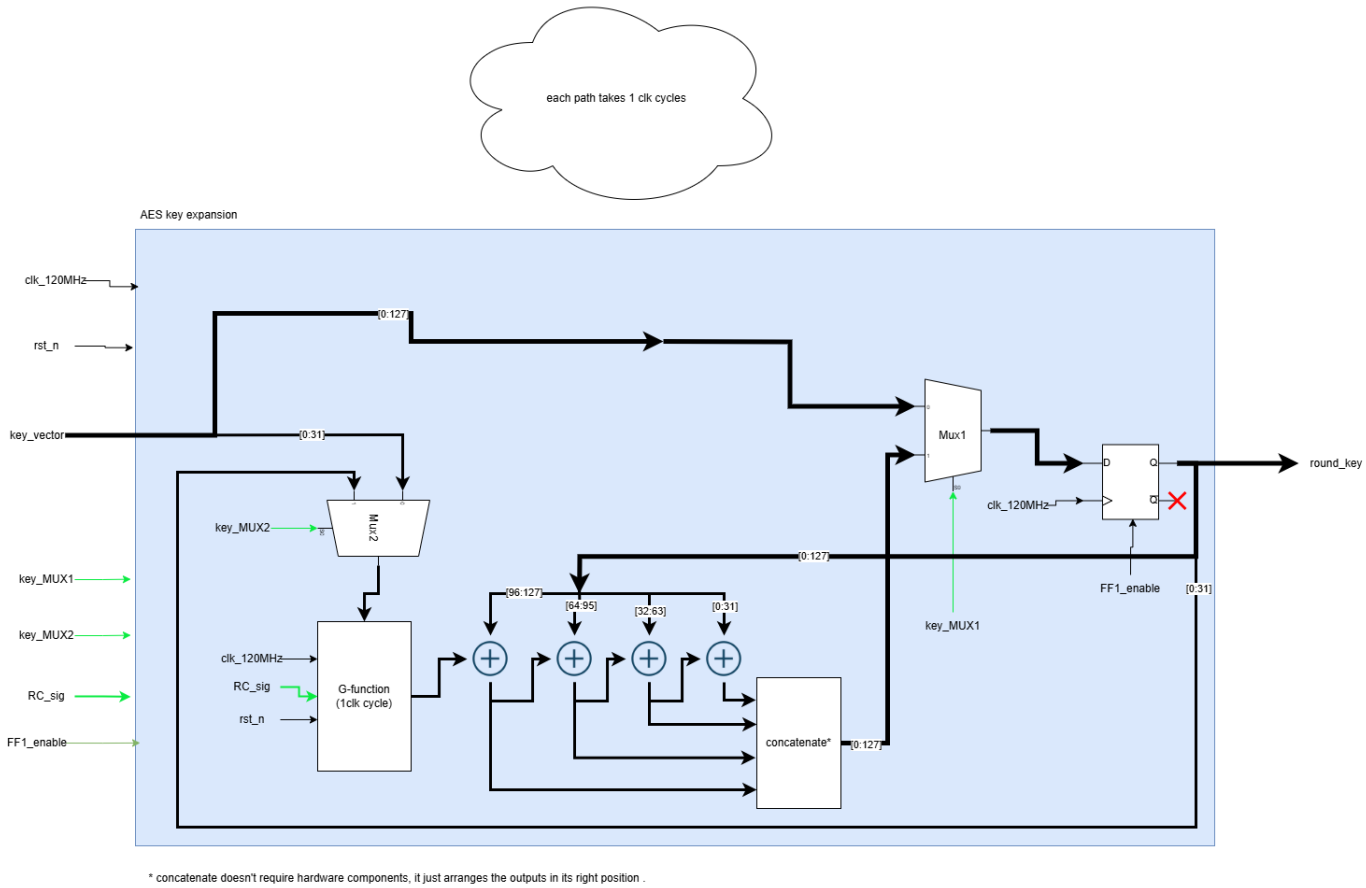
1. Parallel input: which receives all the bits of the AES key and AES input vector in one clock cycle.
2. Serial input: which receives each bit of the AES key and AES input vector in each clock cycle. This requires 128 clock cycles at the beginning of the system operation to form the 128 required bits

We will use the first approach as there is no specification on the system's input format. The used clock frequency will be 120MHz. Latency introduced is 22 clock cycles.

The following section describes the operation of each block of the AES-128 in detail.

AES key expansion

Generates 10 keys. Each key takes 1 clock cycle to be generated .The first-round key is the same as the AES key, thus, we should bypass the first 128 bits to the AES_rounds_operation block. This is done using a 2x1MUX. The rest of the round operations are done using generated round keys. For further details please refer to AES definition document.



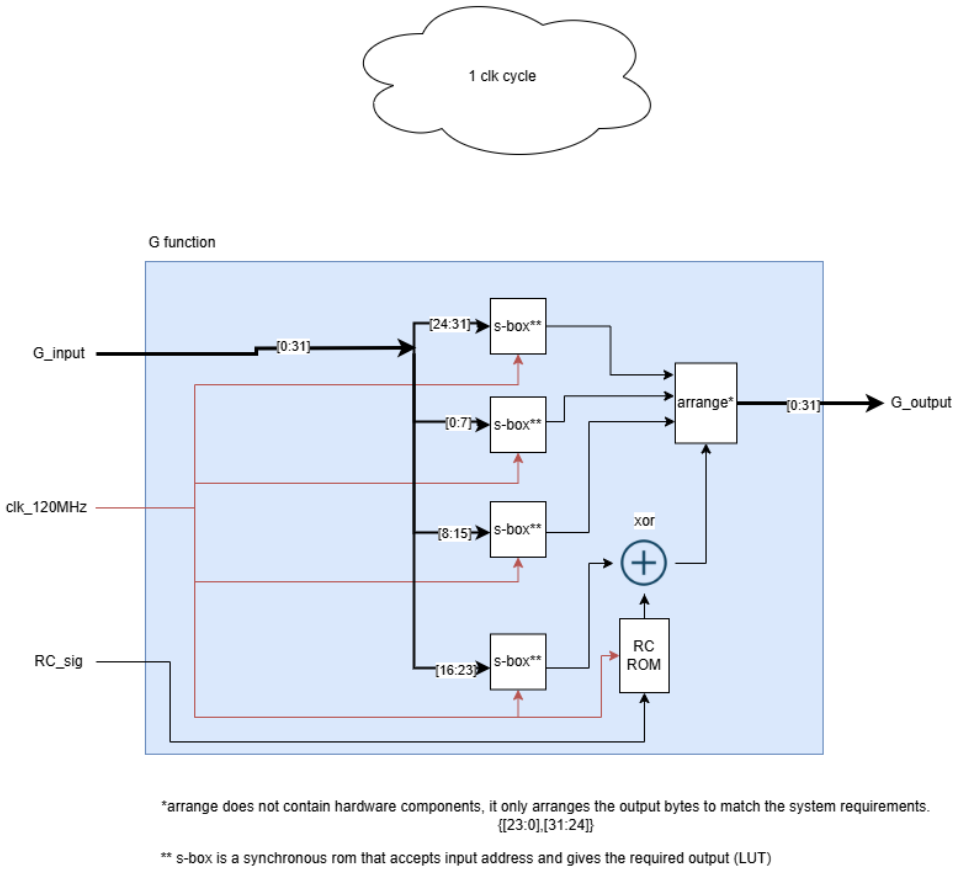
* concatenate doesn't require hardware components, it just arranges the outputs in its right position .

Port Name	Port Width	Port Direction	Description
Clk_120	1	IN	120MHz clock
Rst_n	1	IN	Negative edge reset
Key_MUX1	1	IN	MUX1 select line
Key_MUX2	1	IN	MUX2 select line
FF1_enable	1	IN	Flipflop enable
RC_sig	4	IN	RC ROM address line
Key_vector	128	IN	Initial vector given by user
Round_key	128	OUT	Round keys provided to AES rounds block

G function

Generates the first word of each key based on the previous key's last word. Performs 3 main operations:

1. S box substitution.
2. RC addition.
3. Rearrange of bytes.

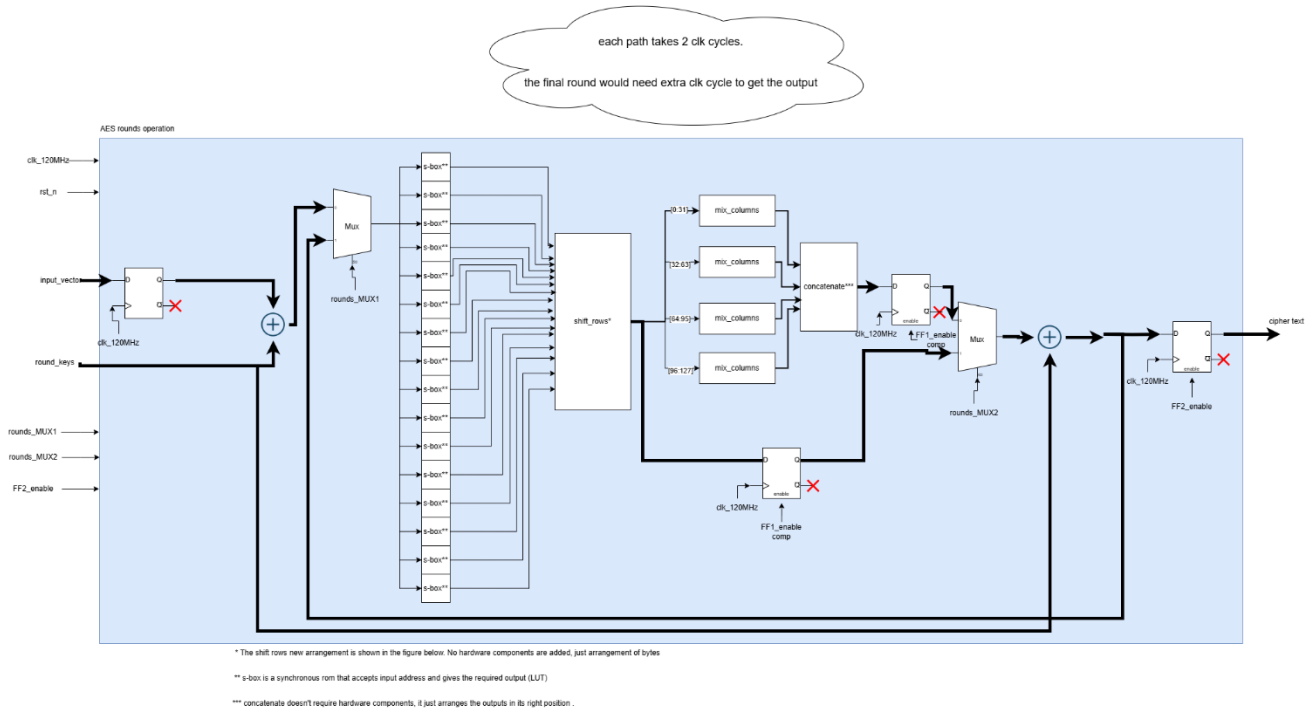


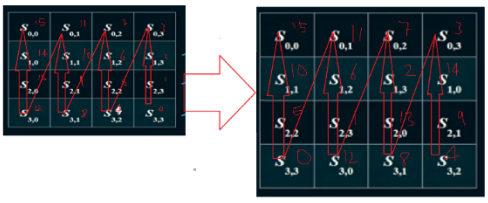
Port Name	Port Width	Port Direction	Description
Clk_120	1	IN	120MHz clock
G_input	32	IN	First word input
RC_sig	4	IN	RC ROM address line
G_output	32	OUT	First key word output

AES rounds operation

Cyclic operations occur depending on each other. Each cycle takes two clock cycles for total of 10 cyclic loops (20 clock cycles) with an extra cycle to store the final output. 4 Operations performed each cycle:

1. S-box substitution.
2. Shift rows.
3. Mix columns.
4. Round key addition.

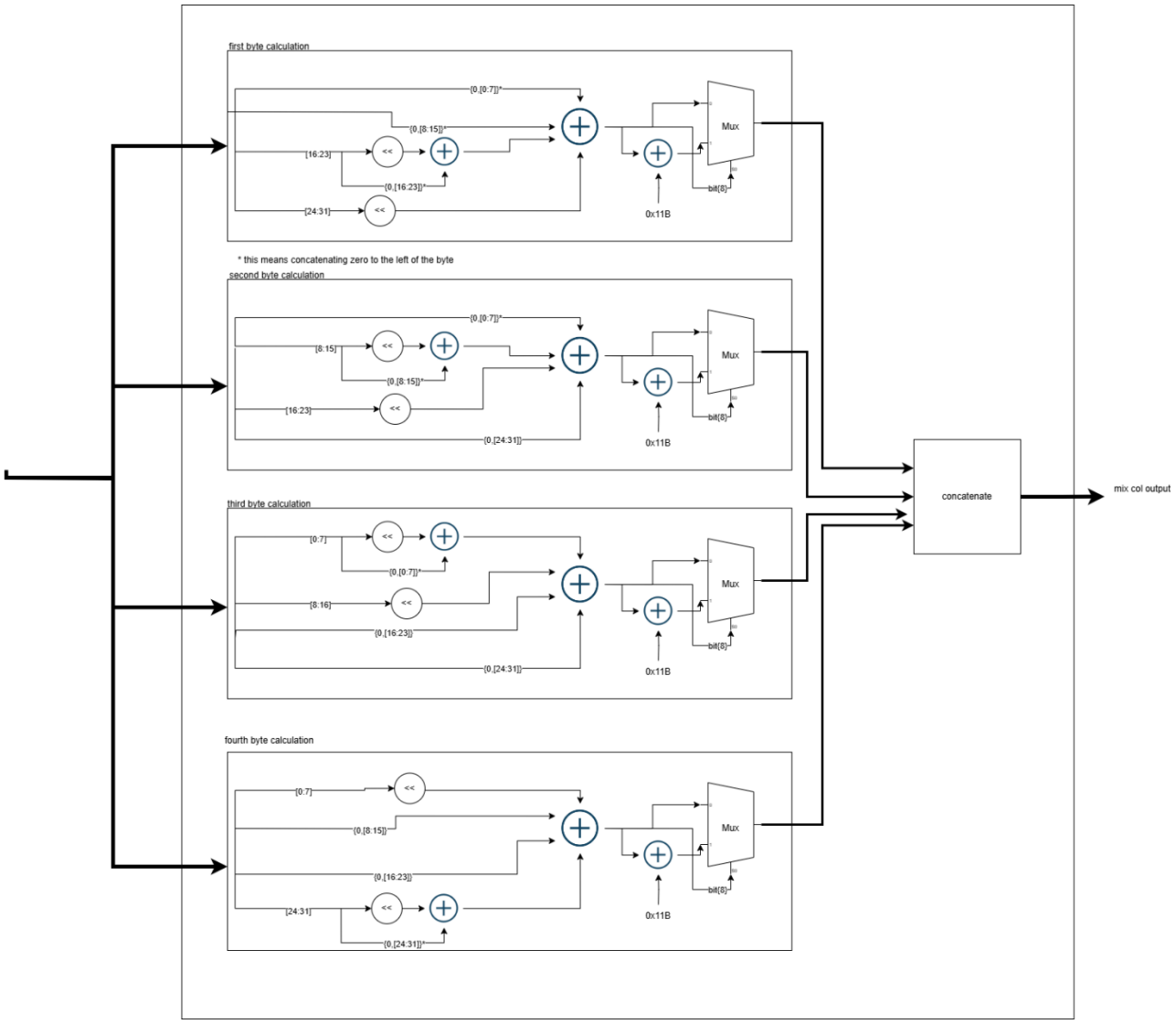




Port Name	Port Width	Port Direction	Description
Clk_120	1	IN	120MHz clock
Rst_n	1	IN	Negative edge reset
Rounds_MUX1	1	IN	MUX1 select line
Rounds_MUX2	1	IN	MUX2 select line
FF1_enable_comp	1	IN	Flipflip 2 enable
FF2_enable	1	IN	Last flipflop enable
input_vector	128	IN	Initial input vector given by user
Round_keys	128	IN	Round keys provided to AES rounds block
Cipher_text	128	OUT	Final encrypted vector

Mix columns

Perform certain invertible equations to the input words (for more details refer to AES definition document).

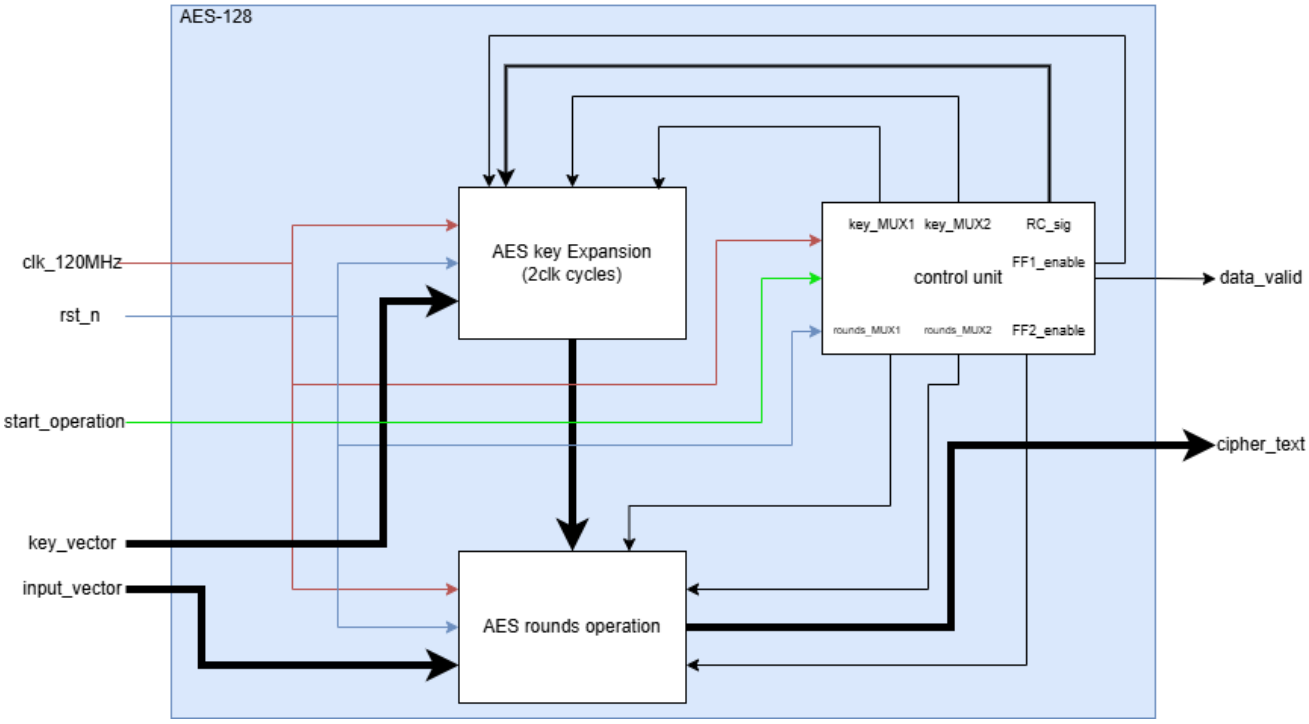


Port Name	Port Width	Port Direction	Description
Mix_columns_in	32	IN	Input word
Mix_columns_out	32	OUT	Output word

Controller unit

Port Name	Port Width	Port Direction	Description
Clk_120	1	IN	120MHz clock
Rst_n	1	IN	Negative edge reset
Start_operation	1	IN	Beginning of system operation
Key_MUX1	1	OUT	MUX1 selection of AES key block
Key_MUX2	1	OUT	MUX2 selection of AES key block
Rounds_MUX1	1	OUT	MUX1 selection of AES rounds block
Rounds_MUX2	1	OUT	MUX2 selection of AES rounds block
FF_key_enable	1	OUT	AES key block flipflop enable
FF_round_enable	1	OUT	AES rounds block flipflop enable
RC_sig	4	OUT	RC ROM address line
Data_valid	1	OUT	End of operation signal

AES 128 wrapper



Port Name	Port Width	Port Direction	Description
Clk_120	1	IN	120MHz clock
Rst_n	1	IN	Negative edge reset
Start_operation	1	IN	Beginning of system operation
Key_vector	128	IN	User input key vector
Input_vector	128	IN	User input vector to be cyphered
Cipher_text	128	OUT	Ciphered text
Data_valid	1	OUT	End of operation signal

➢ FPGA Synthesis

System FPGA implementation was done on ZYNQ Ultrascale+ ZCU104 evaluation board. With the only constraint of 120MHz clock frequency, timing was met.

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	6.526 ns	Worst Hold Slack (WHS):	0.011 ns	Worst Pulse Width Slack (WPWS):	3.624 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	1688	Total Number of Endpoints:	1688	Total Number of Endpoints:	923

All user specified timing constraints are met.

➢ Verification

A UVM based testbench is created to ensure the block's results with respect to MATLAB model results. 200 random input and key vectors were generated to catch if there is mismatch between the model and the RTL. Another 2 testcase were forced where key vector and input vector have same values with all zeros and all ones.

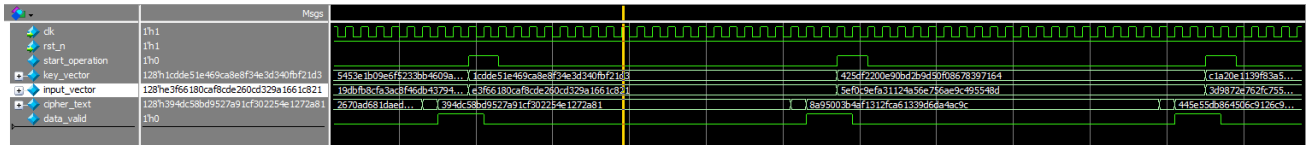
All cases passed based on the following message:

```
# cipher text RTL output: 445e55db864506c9126c915a74d4f10e
# cipher text Model output: 445e55db864506c9126c915a74d4f10e
# data valid output: 1
# The shown parameters are in subscriber
# cipher text RTL output: af8d0977fffd485b14954e6502f7b87b
# cipher text Model output: af8d0977fffd485b14954e6502f7b87b
# data valid output: 1
# The shown parameters are in subscriber
# cipher text RTL output: 66e94bd4ef8a2c3b884cfa59ca342b2e
# cipher text Model output: 66e94bd4ef8a2c3b884cfa59ca342b2e
# data valid output: 1
# The shown parameters are in subscriber
# cipher text RTL output: bcbf217cb280cf30b2517052193ab979
# cipher text Model output: bcbf217cb280cf30b2517052193ab979
# data valid output: 1
# run phase of test is done!
# UVM_INFO verilog_src/uvm-1.1d/src/base/uvm_objection.svh(1267) @ 48486: reporter [TEST_DONE] 'run' phase is ready to proceed to the 'extract' phase
# UVM_INFO uvm_pack_class.sv(379) @ 48486: uvm_test_top.env_in_test.scoreboard_in_env [report_phase] total errors occured:          0
#
```

Code coverage of using previous testcase reached 99.94%.

```
Total Coverage By Instance (filtered view): 99.94%
```

The simulation results can also appear on the waveform using QuestaSim simulation tool.



➢ Future Work

1. Make the system configurable with AES-192 and AES-256.
2. Increase the clock frequency where the system can operate at.
3. Use systemverilog DPI when verifying the results between the hardware and software.