Hardware Assignment 3

Pramod Kumar Meena (2023CS51175) Manvendra Rajpurohit (2023CS10936)

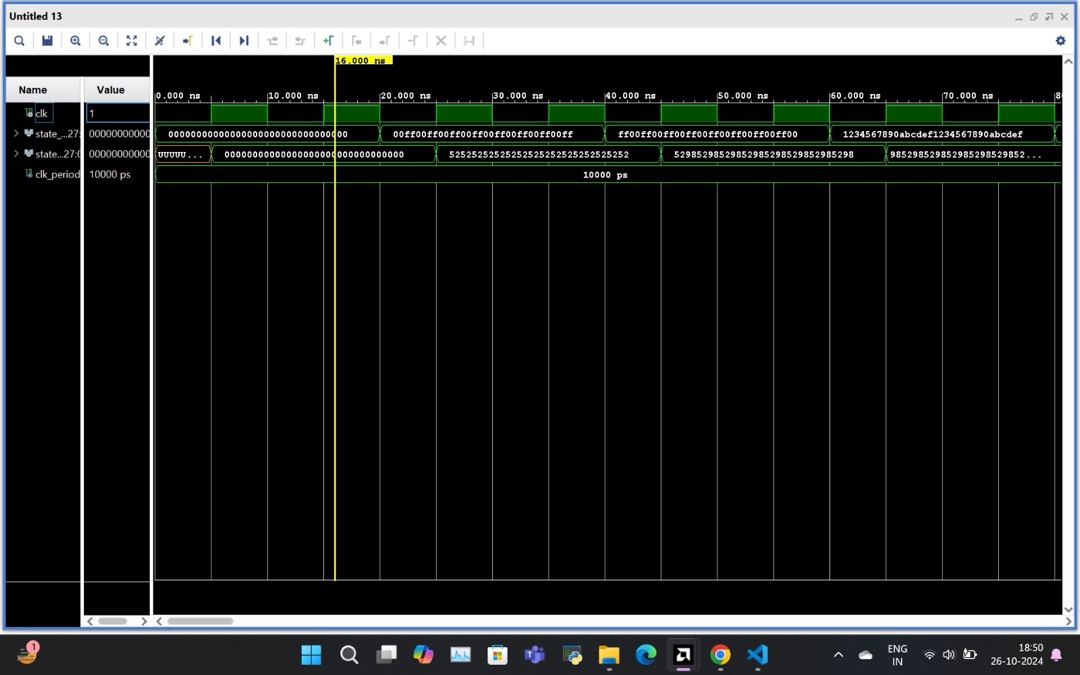
**Problem Statement:**

This project aimed to implement the AES decryption algorithm, including the design of a Finite State Machine (FSM) to control data flow through the system’s memory and computational units. The system uses AES decryption to display decrypted plaintext on a 7-segment display. Key operations, such as InvShiftRows, InvSubBytes, InvMixColumns, and AddRoundKey, are implemented through modular components.

**Approach:**

1. **inv\_sub.vhd**

In this step, each byte is substituted by another byte. It is performed using a lookup table also called the S-box table stored in block memory (blk\_mem\_gen\_box). This substitution is done in a way that a byte is never substituted by itself and also not substituted by another byte which is a compliment of the current byte.

* Clock Input (clk): Synchronizes the data output.
* Component (blk\_mem\_gen\_box): Represents a block memory where the inverse S-box is stored.
* Operation: Each byte of state\_in is sent to the S-box, retrieving the corresponding inverse byte, which is then stored in state\_out\_reg.
* Clocked Process: Updates state\_out with state\_out\_reg on each clock rising edge, ensuring synchronized output.

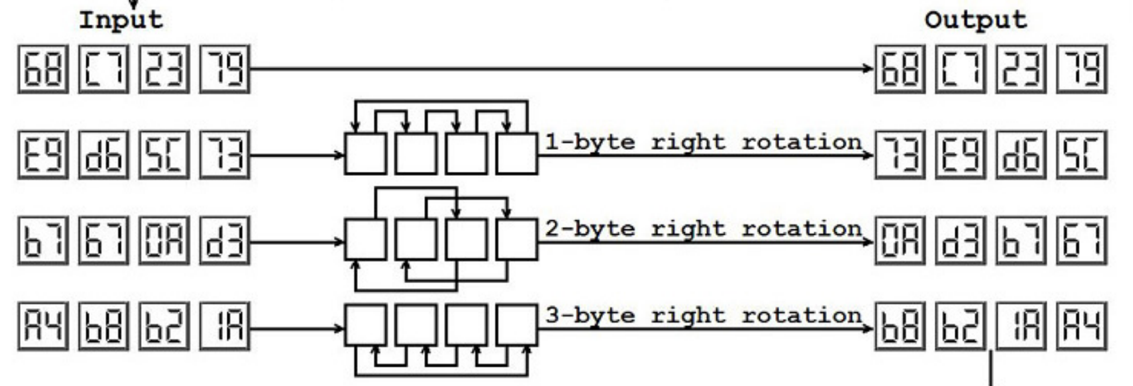
**A screenshot of a computer

Description automatically generated**

**2. inv\_shift\_rows.vhd**

Thimodule does the row-wise shifts. Each row of the 4x4 byte state matrix is shifted right by a specific number of positions (just to reverse the process of encryption).

* Row 0: No shift.
* Row 1: Right shift by 1 byte.
* Row 2: Right shift by 2 bytes.
* Row 3: Right shift by 3 bytes.



A screenshot of a computer

Description automatically generated

A screenshot of a computer program

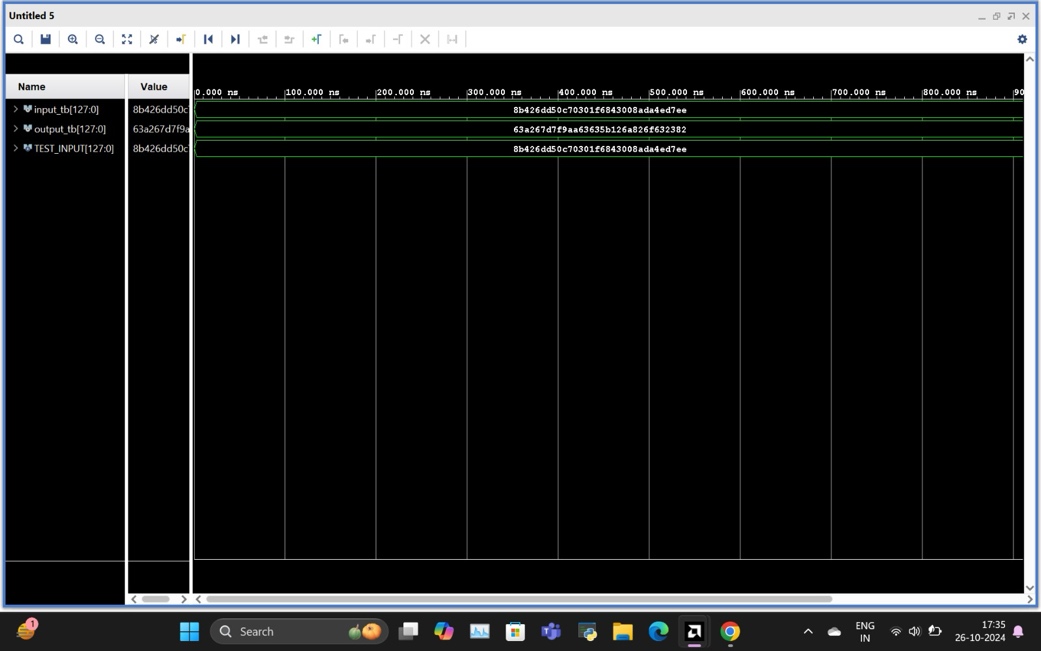
Description automatically generated

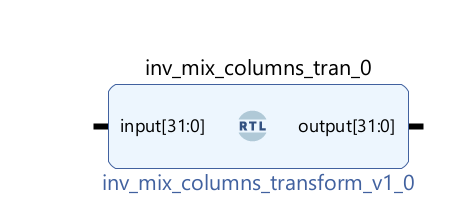
1. **inv\_mix\_columns.vhd**

This module performs the matrix multiplication in the Galois Field GF(2^8) to reverse the column mixing applied during encryption. It takes a 128-bit input vector and applies the inverse mix columns matrix to obtain the output vector.

* Matrix Representation: The 128-bit input is represented as a 4x4 matrix of 8-bit elements using the to\_matrix function, making matrix operations more intuitive.
* Inverse Mix Columns Matrix: Defined as a constant (INV\_MIX\_MATRIX), it contains the fixed values required for decryption.
* Galois Field Multiplication (gf\_mult): A function that handles multiplication within GF(2^8), utilizing shifts and XOR operations to implement finite field arithmetic.
* Matrix Multiplication: For each byte in the resulting matrix, gf\_mult is used to combine the corresponding bytes from INV\_MIX\_MATRIX and state\_matrix.

This module's process block executes the matrix transformation and converts the final 4x4 matrix back into a 128-bit vector with to vector.





**4. add\_round.vhd**

This module performs a bitwise XOR operation between the state and round\_key inputs.

* Bitwise XOR: Each byte of state is XORed with the corresponding byte in round\_key, stored directly in result.
* Clockless Operation: This is a purely combinational logic module, immediately outputting the XOR result without requiring clock synchronization.

The AddRoundKey operation is straightforward yet crucial in reversing the encryption, ensuring that each round’s transformation aligns with the original key schedule.

A screenshot of a computer

Description automatically generated

A screenshot of a computer code

Description automatically generated

**Vivado Memory Implementation**

We used Vivado's Distributed Memory Generator to implement the ROM and RAM. The ROM is initialized with the COE file's contents, while the RAM holds intermediate states. A test bench was created to verify memory functionality, ensuring correct data handling across addresses.

**A screenshot of a computer

Description automatically generated**

**Memory Components**

* blk\_mem\_gen\_2: Cipher memory block
* blk\_mem\_gen\_3: Round memory block

**AES Decryption Process**

Each AES decryption round involves:

* **AddRoundKey**: Bitwise XOR operation with the round key.
* **InvShiftRows**: Right shift each row of the state matrix cyclically.
* **InvSubBytes**: Replace each byte with its corresponding value from the inverse S-box.
* **InvMixColumns**: Reverse the MixColumns operation using multiplication in Galois Field GF(2^8).

**Read Write FSM:**

**A screenshot of a computer

Description automatically generated**

Read write fsm simulations

* **FSM Structure**: Designed to synchronize data transfers and manage the sequence of operations in each AES decryption round.
* **Sequential and Combinational Logic**: State transitions are based on clock signals and flags for operations completion.
* **State Diagram**: Included a state diagram to illustrate transitions for memory read/write and round operations.

**Main FSM** (AES decryption fsm)

**FSM States**

1. **IDLE**: Initializes the state machine. If the start signal is high, loads ciphertext into the state registers.
2. **FIRST\_ROUND**: Executes the initial AddRoundKey operation using the round key.
3. **INV\_SHIFT\_ROWS**: Shifts rows to reverse AES shift transformations.
4. **INV\_SUB\_BYTES**: Substitutes bytes via the inverse S-Box to revert the encryption transformation.
5. **ADD\_ROUND\_KEY**: Combines the current state with the round key.
6. **INV\_MIX\_COLUMNS**: Reverses the MixColumns transformation through a secondary state machine (inv\_mix\_cols\_state) that handles column transformations for each state row.
7. **CHECK\_ROUND**: Decrements round count. If rounds remain, the FSM cycles back to InvShiftRows for further rounds; otherwise, proceeds to FINAL.
8. **FINAL**: Completes decryption, outputs plaintext, and sets the done flag.

**Terms:**

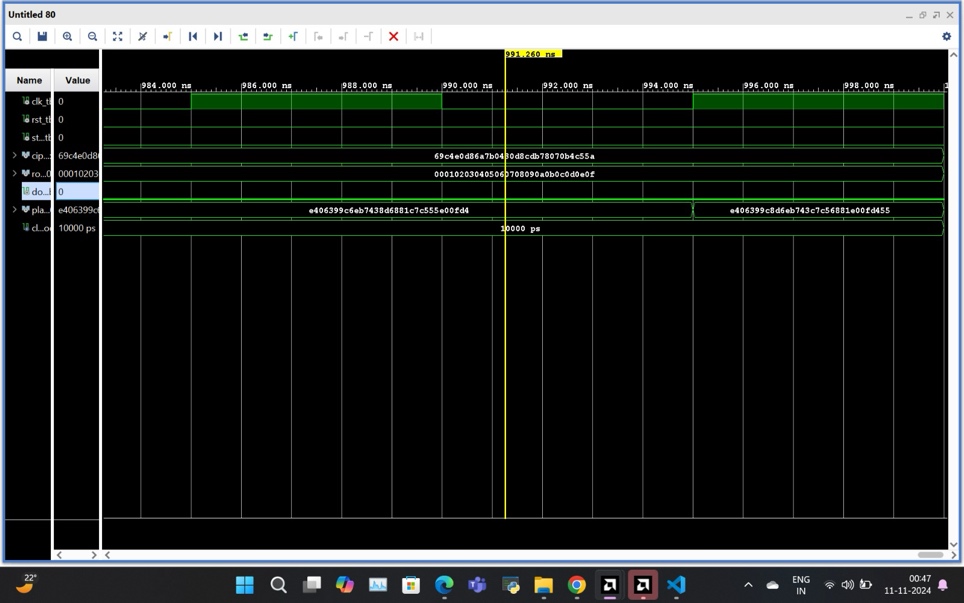
**Sub-modules**: The FSM utilizes separate modules for each AES transformation, including:

* inv\_shift\_rows\_transform for row shifting.
* inverse\_subbytes\_parallel for inverse S-box lookups.
* add\_round for XORing with the round key.
* inv\_mix\_columns\_transform for column transformation.

**Control Signals**: Signals like operation\_done, round\_count, row\_counter, and col\_counter synchronize operations across each state.

**Round Control**: The round\_count signal iterates the FSM over 10 rounds, ensuring compliance with AES decryption standards.

**Wait Cycles**: Minimal wait cycles manage latency, ensuring precise data flow between sub-modules.



A screenshot of a computer

Description automatically generated

**A screenshot of a computer

Description automatically generated**

**A diagram of a computer code

Description automatically generated**

**Display Module (cyclic\_display.vhd)**

* **Plaintext Display**: Final plaintext was converted to hexadecimal characters for display on a 7-segment display.
* **Scrolling Mechanism**: A cyclic scrolling mechanism was implemented to display each character sequentially.

**A screenshot of a computer

Description automatically generated**The plaintext is displayed on the Basys 3 board’s 7-segment display. A 4-character limit is set, with text scrolling in a cyclic manner. Any character outside the displayable ASCII range is shown as “-” by default.

A screen shot of a computer

Description automatically generated

**Test Cases and Results:**

**Summary of Testing and Results**

**Summary:**

The modular implementation of AES decryption provides a highly efficient approach to hardware-based decryption. Each module's functionality was verified through simulations, ensuring accurate decryption of the cipher text. This design not only facilitates testing but also simplifies the integration of components for full-system verification on the Basys 3 board.

**References**

* Basys 3 Reference Manual
* IEEE VHDL Reference Manual
* ASCII Table for Plaintext Conversion

**Report Utilisation File:**

Copyright 1986-2022 Xilinx, Inc. All Rights Reserved. Copyright 2022-2024 Advanced Micro Devices, Inc. All Rights Reserved.

---------------------------------------------------------------------------------------------------------------------------------------------

| Tool Version : Vivado v.2024.1 (win64) Build 5076996 Wed May 22 18:37:14 MDT 2024

| Date : Mon Nov 11 01:30:46 2024

| Host : Manvendrarajpur running 64-bit major release (build 9200)

| Command : report\_utilization -file read\_write\_fsm\_utilization\_synth.rpt -pb read\_write\_fsm\_utilization\_synth.pb

| Design : read\_write\_fsm

| Device : xc7a35tcpg236-1

| Speed File : -1

| Design State : Synthesized

---------------------------------------------------------------------------------------------------------------------------------------------

Utilization Design Information

Table of Contents

-----------------

1. Slice Logic

1.1 Summary of Registers by Type

2. Memory

3. DSP

4. IO and GT Specific

5. Clocking

6. Specific Feature

7. Primitives

8. Black Boxes

9. Instantiated Netlists

1. Slice Logic

--------------

+-------------------------+------+-------+------------+-----------+-------+

| Site Type | Used | Fixed | Prohibited | Available | Util% |

+-------------------------+------+-------+------------+-----------+-------+

| Slice LUTs\* | 78 | 0 | 0 | 20800 | 0.38 |

| LUT as Logic | 78 | 0 | 0 | 20800 | 0.38 |

| LUT as Memory | 0 | 0 | 0 | 9600 | 0.00 |

| Slice Registers | 277 | 0 | 0 | 41600 | 0.67 |

| Register as Flip Flop | 265 | 0 | 0 | 41600 | 0.64 |

| Register as Latch | 12 | 0 | 0 | 41600 | 0.03 |

| F7 Muxes | 0 | 0 | 0 | 16300 | 0.00 |

| F8 Muxes | 0 | 0 | 0 | 8150 | 0.00 |

+-------------------------+------+-------+------------+-----------+-------+

\* Warning! The Final LUT count, after physical optimizations and full implementation, is typically lower. Run opt\_design after synthesis, if not already completed, for a more realistic count.

Warning! LUT value is adjusted to account for LUT combining.

Warning! For any ECO changes, please run place\_design if there are unplaced instances

1.1 Summary of Registers by Type

--------------------------------

+-------+--------------+-------------+--------------+

| Total | Clock Enable | Synchronous | Asynchronous |

+-------+--------------+-------------+--------------+

| 0 | \_ | - | - |

| 0 | \_ | - | Set |

| 0 | \_ | - | Reset |

| 0 | \_ | Set | - |

| 0 | \_ | Reset | - |

| 0 | Yes | - | - |

| 0 | Yes | - | Set |

| 277 | Yes | - | Reset |

| 0 | Yes | Set | - |

| 0 | Yes | Reset | - |

+-------+--------------+-------------+--------------+

2. Memory

---------

+-------------------+------+-------+------------+-----------+-------+

| Site Type | Used | Fixed | Prohibited | Available | Util% |

+-------------------+------+-------+------------+-----------+-------+

| Block RAM Tile | 1 | 0 | 0 | 50 | 2.00 |

| RAMB36/FIFO\* | 0 | 0 | 0 | 50 | 0.00 |

| RAMB18 | 2 | 0 | 0 | 100 | 2.00 |

| RAMB18E1 only | 2 | | | | |

+-------------------+------+-------+------------+-----------+-------+

\* Note: Each Block RAM Tile only has one FIFO logic available and therefore can accommodate only one FIFO36E1 or one FIFO18E1. However, if a FIFO18E1 occupies a Block RAM Tile, that tile can still accommodate a RAMB18E1

3. DSP

------

+-----------+------+-------+------------+-----------+-------+

| Site Type | Used | Fixed | Prohibited | Available | Util% |

+-----------+------+-------+------------+-----------+-------+

| DSPs | 0 | 0 | 0 | 90 | 0.00 |

+-----------+------+-------+------------+-----------+-------+

4. IO and GT Specific

---------------------

+-----------------------------+------+-------+------------+-----------+--------+

| Site Type | Used | Fixed | Prohibited | Available | Util% |

+-----------------------------+------+-------+------------+-----------+--------+

| Bonded IOB | 264 | 0 | 0 | 106 | 249.06 |

| Bonded IPADs | 0 | 0 | 0 | 10 | 0.00 |

| Bonded OPADs | 0 | 0 | 0 | 4 | 0.00 |

| PHY\_CONTROL | 0 | 0 | 0 | 5 | 0.00 |

| PHASER\_REF | 0 | 0 | 0 | 5 | 0.00 |

| OUT\_FIFO | 0 | 0 | 0 | 20 | 0.00 |

| IN\_FIFO | 0 | 0 | 0 | 20 | 0.00 |

| IDELAYCTRL | 0 | 0 | 0 | 5 | 0.00 |

| IBUFDS | 0 | 0 | 0 | 104 | 0.00 |

| GTPE2\_CHANNEL | 0 | 0 | 0 | 2 | 0.00 |

| PHASER\_OUT/PHASER\_OUT\_PHY | 0 | 0 | 0 | 20 | 0.00 |

| PHASER\_IN/PHASER\_IN\_PHY | 0 | 0 | 0 | 20 | 0.00 |

| IDELAYE2/IDELAYE2\_FINEDELAY | 0 | 0 | 0 | 250 | 0.00 |

| IBUFDS\_GTE2 | 0 | 0 | 0 | 2 | 0.00 |

| ILOGIC | 0 | 0 | 0 | 106 | 0.00 |

| OLOGIC | 0 | 0 | 0 | 106 | 0.00 |

+-----------------------------+------+-------+------------+-----------+--------+

5. Clocking

-----------

+------------+------+-------+------------+-----------+-------+

| Site Type | Used | Fixed | Prohibited | Available | Util% |

+------------+------+-------+------------+-----------+-------+

| BUFGCTRL | 1 | 0 | 0 | 32 | 3.13 |

| BUFIO | 0 | 0 | 0 | 20 | 0.00 |

| MMCME2\_ADV | 0 | 0 | 0 | 5 | 0.00 |

| PLLE2\_ADV | 0 | 0 | 0 | 5 | 0.00 |

| BUFMRCE | 0 | 0 | 0 | 10 | 0.00 |

| BUFHCE | 0 | 0 | 0 | 72 | 0.00 |

| BUFR | 0 | 0 | 0 | 20 | 0.00 |

+------------+------+-------+------------+-----------+-------+

6. Specific Feature

-------------------

+-------------+------+-------+------------+-----------+-------+

| Site Type | Used | Fixed | Prohibited | Available | Util% |

+-------------+------+-------+------------+-----------+-------+

| BSCANE2 | 0 | 0 | 0 | 4 | 0.00 |

| CAPTUREE2 | 0 | 0 | 0 | 1 | 0.00 |

| DNA\_PORT | 0 | 0 | 0 | 1 | 0.00 |

| EFUSE\_USR | 0 | 0 | 0 | 1 | 0.00 |

| FRAME\_ECCE2 | 0 | 0 | 0 | 1 | 0.00 |

| ICAPE2 | 0 | 0 | 0 | 2 | 0.00 |

| PCIE\_2\_1 | 0 | 0 | 0 | 1 | 0.00 |

| STARTUPE2 | 0 | 0 | 0 | 1 | 0.00 |

| XADC | 0 | 0 | 0 | 1 | 0.00 |

+-------------+------+-------+------------+-----------+-------+

7. Primitives

-------------

+----------+------+---------------------+

| Ref Name | Used | Functional Category |

+----------+------+---------------------+

| FDCE | 265 | Flop & Latch |

| OBUF | 257 | IO |

| LUT4 | 70 | LUT |

| LUT6 | 36 | LUT |

| LDCE | 12 | Flop & Latch |

| LUT2 | 7 | LUT |

| IBUF | 7 | IO |

| RAMB18E1 | 2 | Block Memory |

| LUT5 | 2 | LUT |

| BUFG | 1 | Clock |

+----------+------+---------------------+

8. Black Boxes

--------------

+----------+------+

| Ref Name | Used |

+----------+------+

9. Instantiated Netlists

------------------------

+----------+------+

| Ref Name | Used |

+----------+------+