CND 221: Advanced Full Custom Design

Final Project

Simplified-Data Encryption standard (S-DES)

Submitted by

Student Name	Section	ID
Fagr Ahmed	12	V23010586
Shrouq El Shaal	12	V23010529
Marwa Moaz	14	V23009897
Mohamed Yasser	16	V23010495

1.Abstract
2.Introduction
3. DES
4. Simplified-DES
5. System functionality
    - Encryption
        - IP
        - FK
            A. EP
            B. P4
            C. S-Box(S0,S1)
        - SW
        - IP-1
    - Decryption
    - Keys
        - P10
        - P8
        - LS1
        - LS2
6. Behavioral stage
    - RTL
    - simulation
    - estimate the sizes of various components of design (eg. number of gates or transistors)
7. schematic and cell designs for each block
8. Final simulation results (individual blocks and integrated system)
9.Completed the layout of all blocks and started routing and placement
    - (DRC and LVS).
    - Run Parasitic extraction, back annotate and re-simulate, assess the impact of parasitics on your design.
    - Calculate total Area, power and throughput.
    - Compare actual area to estimated area
10. Conclusion
11. References

1. Abstract:

This paper presents a novel approach to the design and implementation of the Data Encryption Standard (DES) through a full-custom methodology, adhering to a structured design process informed by principles discussed in the course. Following a systematic framework, the project begins with system specification, delineating the functional requirements and architectural constraints of the custom DES implementation.

Leveraging a hierarchical structure of subsystems, each component is developed and verified incrementally, ensuring the creation of robust and interoperable modules. Throughout the design process, emphasis is placed on the application of design methodologies elucidated in the Full custom course, including abstraction, modularity, and design reuse.

Verification procedures are meticulously applied at each stage, guaranteeing the integrity and functionality of individual subsystems as well as their seamless integration into the overarching DES framework. By adhering to this rigorous design approach, the project not only yields a bespoke DES implementation tailored to optimize performance and resource utilization but also serves as a testament to the efficacy of structured design methodologies in cryptographic engineering.

2. Introduction:

In the realm of information security, the Data Encryption Standard (DES) has long stood as a cornerstone, offering a robust framework for safeguarding sensitive data. As the digital landscape continues to evolve, the imperative to innovate and optimize encryption methodologies becomes increasingly pronounced. This paper embarks on a journey to explore a novel approach to DES design and implementation, one that integrates principles discussed in our course on design methodologies.

The project outlined herein follows a structured design process, commencing with meticulous system specification to delineate the functional requirements and architectural constraints of the custom DES implementation. Building upon this foundation, a hierarchical structure of subsystems is employed, allowing for incremental development and testing. This approach not only ensures the creation of working subsystems but also facilitates seamless integration into the overarching DES framework.

Throughout the design endeavor, paramount importance is placed on the application of design methodologies discussed in the course. Concepts such as abstraction, modularity, and design reuse serve as guiding principles, shaping the architecture and implementation of each subsystem. Furthermore, rigorous verification procedures are systematically applied at each stage, guaranteeing the integrity and functionality of the custom DES implementation.

By adhering to this structured design approach, this project aims not only to deliver a bespoke DES implementation optimized for performance and resource utilization but also to underscore the efficacy of structured design methodologies in cryptographic engineering. Through the exploration of this innovative approach, we seek to contribute to the ongoing discourse surrounding encryption methodologies and their role in fortifying information security in an ever-evolving digital landscape.

3. DES:

Data Encryption Standard (DES) is a block cipher with a 56-bit key length that has played a significant role in data security. Data encryption standard (DES) has been found vulnerable to very powerful attacks therefore, the popularity of DES has been found slightly on the decline. DES is a block cipher and encrypts data in blocks of size of 64 bits each, which means 64 bits of plain text go as the input to DES, which produces 64 bits of ciphertext. The same algorithm and key are used for encryption and decryption, with minor differences. The key length is 56 bits.

The basic idea is shown below:

We have mentioned that DES uses a 56-bit key. Actually, The initial key consists of 64 bits. However, before the DES process even starts, every 8th bit of the key is discarded to produce a 56-bit key. That is, bit positions 8, 16, 24, 32, 40, 48, 56, and 64 are discarded.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64

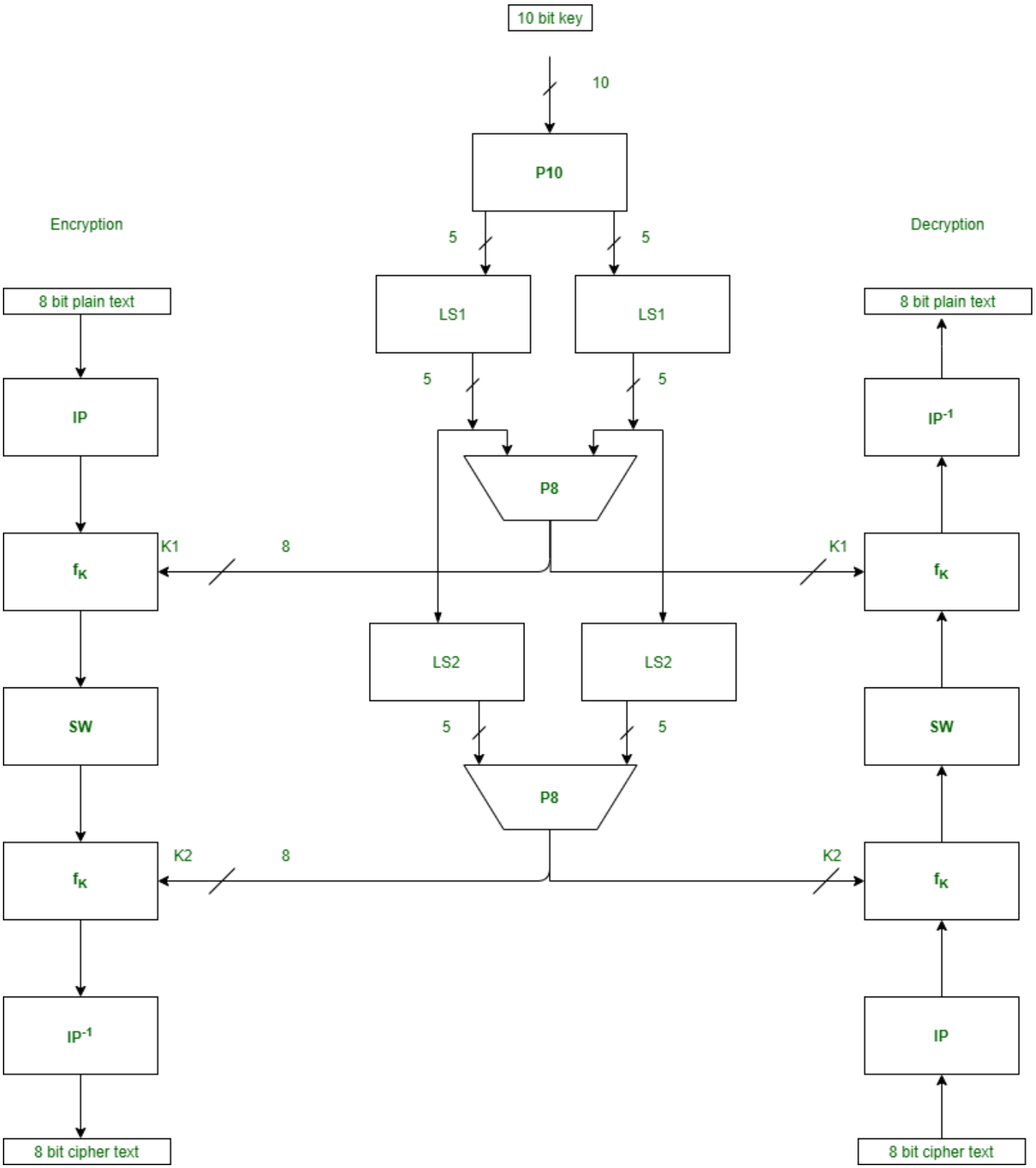
Thus, the discarding of every 8th bit of the key produces a 56-bit key from the original 64-bit key.
DES is based on the two fundamental attributes of cryptography: substitution (also called confusion) and transposition (also called diffusion). DES consists of 16 steps, each of which is called a round. Each round performs the steps of substitution and transposition. Let us now discuss the broad-level steps in DES.

- In the first step, the 64-bit plain text block is handed over to an initial Permutation (IP) function.
- The initial permutation is performed on plain text.
- Next, the initial permutation (IP) produces two halves of the permuted block; saying Left Plain Text (LPT) and Right Plain Text (RPT).
- Now each LPT and RPT go through 16 rounds of the encryption process.
- In the end, LPT and RPT are rejoined and a Final Permutation (FP) is performed on the combined block
- The result of this process produces 64-bit ciphertext.

4. Simplified-DES

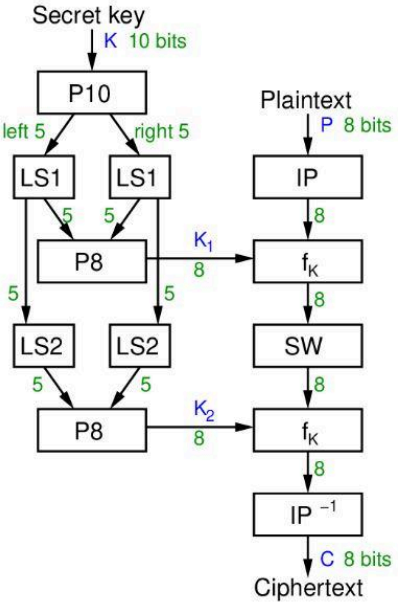
To understand the details of a cipher, it often helps if you can perform the encryption (or decryption) steps yourself. However as common block ciphers operate on blocks of 64 bits or larger, and use similar sized keys, it is difficult to manually and efficiently perform operations. Therefore, to illustrate the principles of selected real ciphers, simplified versions have been developed. This section presents Simplified Data Encryption Standard (S-DES), which is a cut-down version of DES. For example, S-DES operates on 8-bit blocks, uses an 8-bit key and has only 2 rounds. As it is designed using the same principles as (real) DES but using smaller values, it is possible to step through an example encryption by hand. For some this can be a powerful way to understand the operations used in real DES. It is important however to note that S-DES is just for education; it is not a real cipher used in practice today or in the past. You will only find it referred to in textbooks and university classes.

- Input (plaintext) block: 8-bits
- Output (ciphertext) block: 8-bits
- Key: 10-bits
- Rounds: 2
- Round keys generated using permutations and left shifts
- Encryption: initial permutation, round function, switch halves
- Decryption: Same as encryption, except round keys used in opposite order


10 bit key
10
P10
Encryption
Decryption
5
5
LS1
LS1
8 bit plain text
8 bit plain text
5
5
IP
IP-1
P8
K1
8
K1
fK
fK
LS2
LS2
SW
SW
5
5
P8
K2
8
K2
fK
fK
IP-1
IP
8 bit cipher text
8 bit cipher text

5. System functionality:

Encryption: The system accepts plaintext input and performs encryption using the DES algorithm. This process involves breaking the plaintext into fixed-size blocks, applying permutation and substitution operations specified by the DES algorithm, and generating ciphertext as output.Here are the key details about encryption blocks in DES:



1-Block Size: The plaintext is divided into fixed-size blocks, each consisting of 8 bits.

2-Key Size: The key used for encryption and decryption is 10 bits long.

3- Padding: Since the block size is fixed at 8 bits, padding may not be necessary unless dealing with messages that aren't exact multiples of 8 bits.

4- Round Function: In each round of DES, a portion of the key is used to modify the plaintext block. Given the smaller key size, modifications to the round function would be needed to accommodate the 10-bit key.

5- Permutations: DES involves several permutations of the plaintext and the key. These permutations would need to be adjusted to fit the smaller block and key sizes.

6- Expansion Permutation: In DES, an expansion permutation is applied to the plaintext block to increase its size before mixing it with the key. This step would need to be adjusted for the 8-bit block size.

7- S-Boxes: DES uses a series of substitution boxes (S-boxes) to further obfuscate the data. These would need to be adapted for the smaller block size.

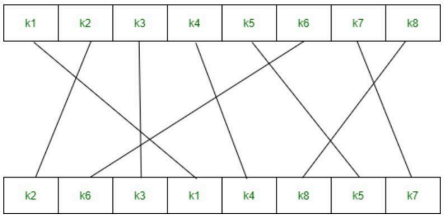
8- Key Schedule: The key schedule algorithm, which generates round keys from the main key, would need to be adjusted to handle the smaller key size.

9- Rounds: DES typically consists of 16 rounds. Each round involves a combination of permutation, substitution, and XOR operations.

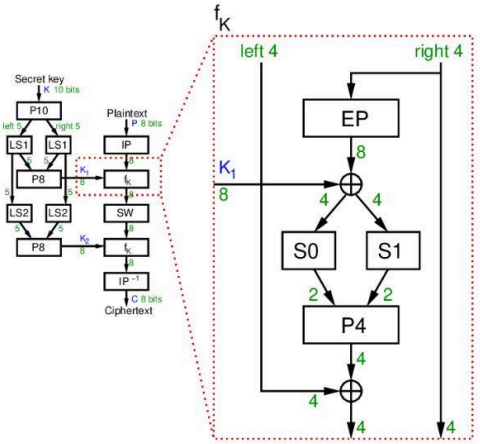
10- Final Permutation: After the last round, a final permutation is applied to the data to generate the ciphertext.

The encryption algorithm involves to four main functions:

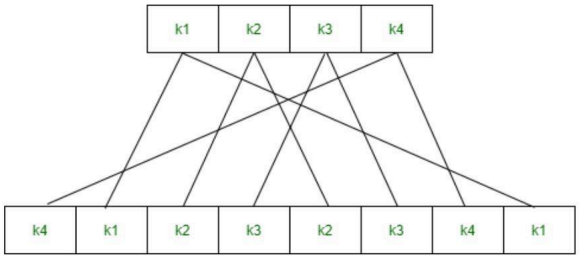
1-**IP:**This step involves rearranging the bits of the plaintext according to a predefined permutation table. The purpose of this step is to diffuse the input bits to provide a good starting point for subsequent operations.



2-**FK**:In this step, the subkey derived from the main encryption key is combined with the input data. This could involve XORing the subkey with a portion of the data or some other operation depending on the specific algorithm.



A. **EP**:It takes a 4-bit input and converts it into an 8-bit output.



B. **S-Box(S0,S1)**: input used to select row/column; selected element is output 4-bit
input: $bit1, bit2, bit3, bit4$
- $bit1bit4$ specifies row (0, 1, 2 or 3 in decimal)
- $bit2bit3$ specifies column

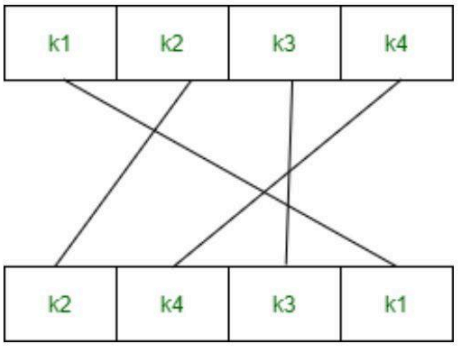
S0

1	0	3	2
3	2	1	0
0	2	1	3
3	1	3	2

S1

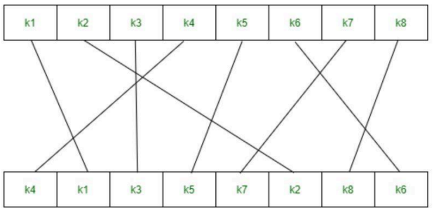
0	1	2	3
2	0	1	3
3	0	1	0
2	1	0	3

C. P4



3-**SW:**This is the primary operation of the encryption process. It typically consists of multiple rounds where a combination of substitution (replacing plaintext bits with other bits) and permutation (rearranging the order of bits) operations are applied iteratively. This step ensures confusion and diffusion of the input data, making it difficult for an attacker to discern any patterns.

4-**IP-1**:This is the final permutation step, which is the inverse of the initial permutation. It rearranges the bits of the output from the substitution-permutation network to produce the final ciphertext.



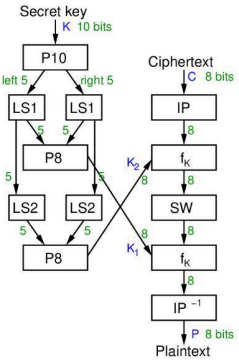
We can concisely express the encryption algorithm as a composition of functions:

$$\text{Encryption Algorithm} = IP \circ FK \circ SW \circ IP^{-1}$$

Where:

IP : represents the Initial Permutation function,
FK :represents the Key Mixing function,
SW: represents the Substitution-Permutation Network function, and
IP^{-1}: represents the Inverse Permutation function.

Decryption in DES (Data Encryption Standard) involves reversing the steps of encryption. DES is a symmetric key block cipher, meaning the same key is used for both encryption and decryption. The decryption process involves using the same key in reverse order to transform the ciphertext back into plaintext.



We can concisely express the Decryption algorithm as a composition of functions:

$$\text{Decryption Algorithm} = IP^{-1} \circ SW \circ FK \circ IP$$

By applying these inverse functions in reverse order, the ciphertext can be transformed back into plaintext using the same key that was used for encryption.

- **Key Generation**

Involves creating the secret key used in the DES algorithm for encrypting and decrypting data. In a simplified sense, for a 10-bit key, the key generation process would generate a set of subkeys needed for encryption and decryption. However, it's important to note that DES typically requires a 56-bit key, not a 10-bit key, for proper encryption. If you're referring to a scenario where you're working with a 10-bit key for educational purposes or a specific application, the key generation process would be adjusted accordingly, likely involving expansion or adaptation to fit the requirements of the DES algorithm with a shorter key length.

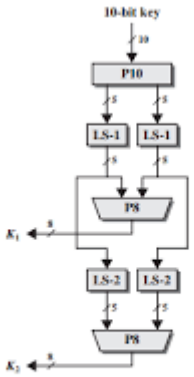


Figure G.2 Key Generation for Simplified DES

- **P10**

Refers to a permutation operation called the "10-bit permutation" that is applied to the original 10-bit key to produce a modified key. This permutation rearranges the bits of the original key according to a specific predefined pattern. The purpose of the P10 permutation is to enhance the security and randomness of the key by spreading the bits across different positions in the key. This modified key is then used in subsequent rounds of the DES algorithm for encryption and decryption.

◆ P8

Refers to a permutation operation called the "8-bit permutation" that is applied to the 10-bit key generated after the initial permutation (P10) and key splitting. This permutation rearranges the bits of the 10-bit key according to a specific predefined pattern, selecting only 8 out of the 10 bits and discarding the remaining 2 bits. The purpose of the P8 permutation is to further modify the key and reduce its size to 8 bits, which will be used as the first subkey in subsequent rounds of the DES algorithm. This subkey generation process helps ensure the security and effectiveness of the encryption process.

◆ LS1

Refers to a process in the key schedule of DES where the bits of the key are shifted to the left by one position. Specifically, in the key schedule, after generating the initial subkeys, each subkey undergoes a left circular shift by one bit position. This shift operation is applied independently to both halves of the 10-bit key, resulting in two modified 5-bit halves. The LS1 operation is repeated in various rounds of the key schedule to generate different subkeys used in the encryption and decryption process. This shifting process helps introduce variability and randomness into the subkeys, enhancing the security of the DES algorithm.

◆ LS2

It is a step in the key schedule of DES where the bits of the key are shifted to the left by two positions. Specifically, after generating the initial subkeys, each subkey undergoes a left circular shift by two bit positions. This shift operation is applied independently to both halves of the 10-bit key, resulting in two modified 5-bit halves. The LS2 operation is repeated in various rounds of the key schedule to generate different subkeys used in the encryption and decryption process. Similar to LS1, this shifting process helps introduce variability and randomness into the subkeys, thereby enhancing the security of the DES algorithm.

6. Behavioral stage

- RTL

In the context of the Data Encryption Standard (DES), RTL typically refers to "Round Transformation Layer."

DES encryption involves multiple rounds of transformations applied to the plaintext data using a series of subkeys derived from the main key. Each round consists of several operations, including permutation, substitution, and shifting. The Round Transformation Layer encompasses these operations within each round of the DES algorithm.

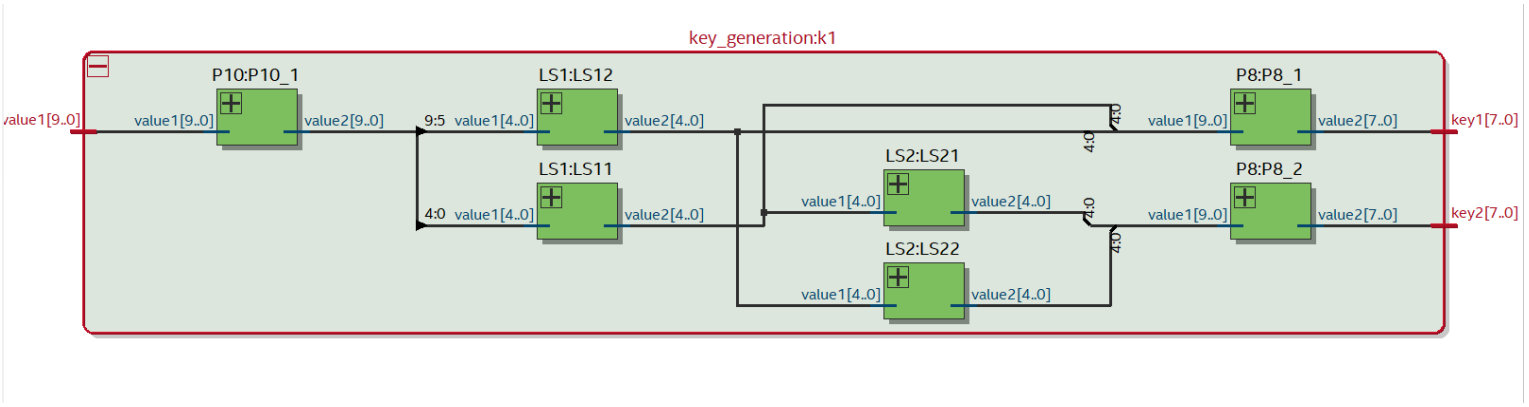
During each round, the RTL performs the following main operations:

1. Expansion Permutation (E): Expand the 8-bit data block to a larger size, possibly to 12 or 16 bits, using a predefined expansion table.
2. XOR with Subkey: XOR the expanded data with a subkey generated for the current round. This subkey might also be 8 bits in this simplified version.
3. Substitution (S-boxes): Divide the XOR result into smaller blocks (e.g., 4 bits each) and substitute each block using S-boxes. Since we're dealing with an 8-bit block size, the number of S-boxes and their sizes might be adjusted accordingly.
4. Permutation (P): Permute the output of the S-box substitution using a fixed permutation table. This step may involve rearranging the bits within the 8-bit block.
5. XOR with the previous left half: Finally, XOR the permuted result with the left half of the previous round's data.

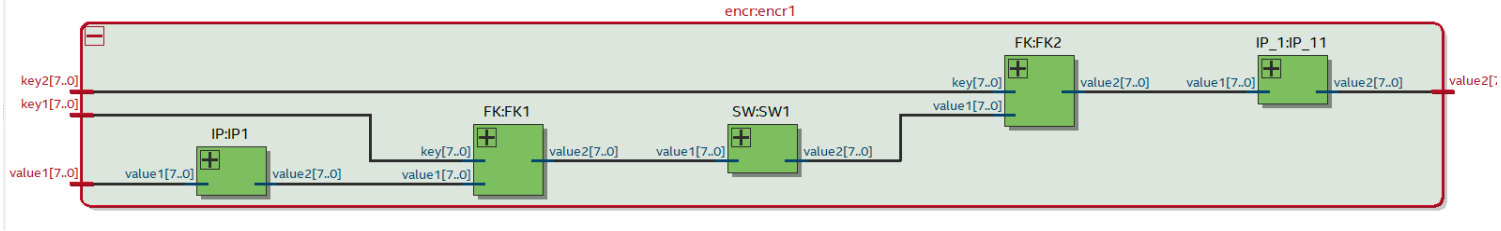
These operations are repeated for the designated number of rounds in the DES algorithm, resulting in the encrypted ciphertext. The Round Transformation Layer is fundamental to the DES encryption process, providing the cryptographic strength and confusion necessary to secure the data.

- simulation

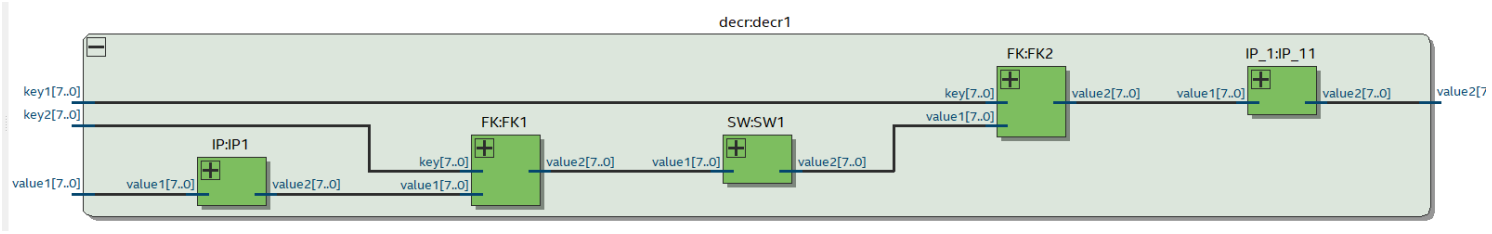
1-Key Generation



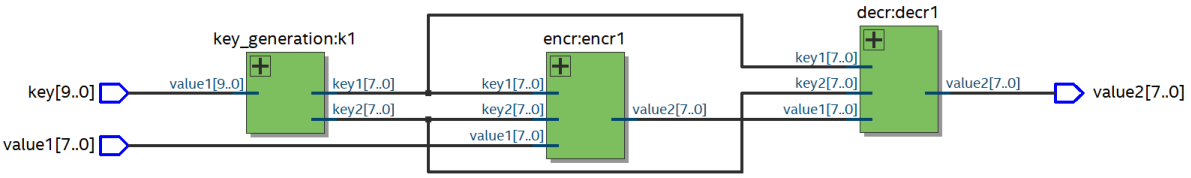
2- Encryption

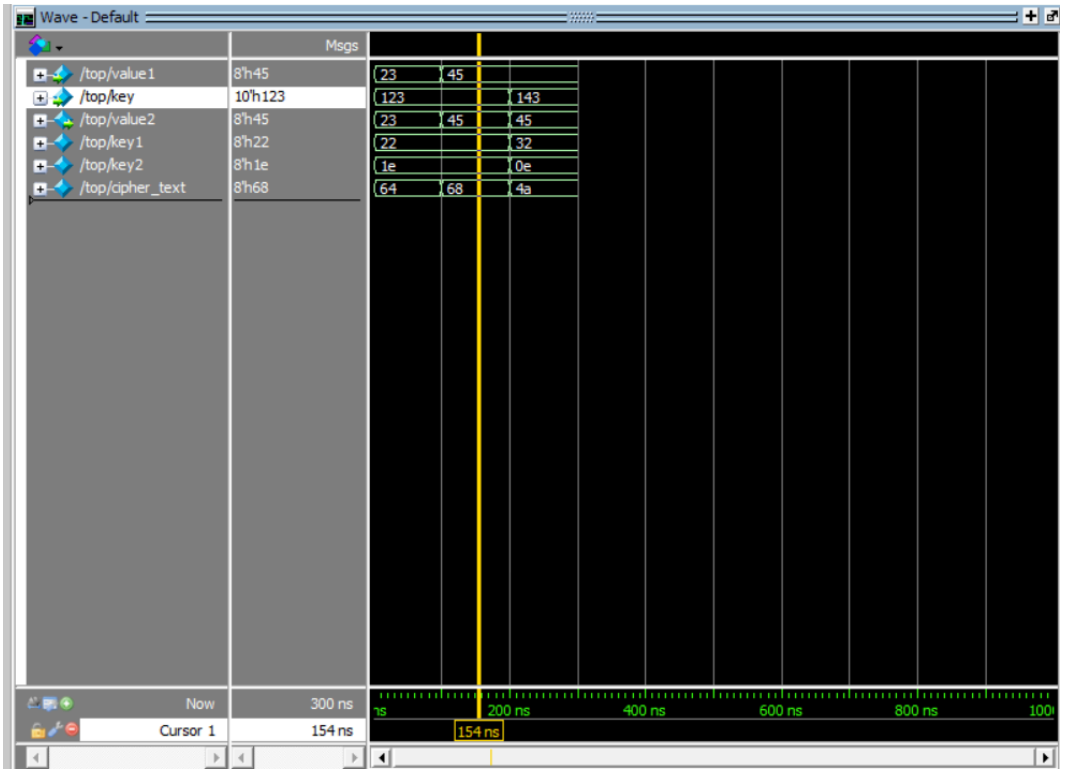


3- Decryption

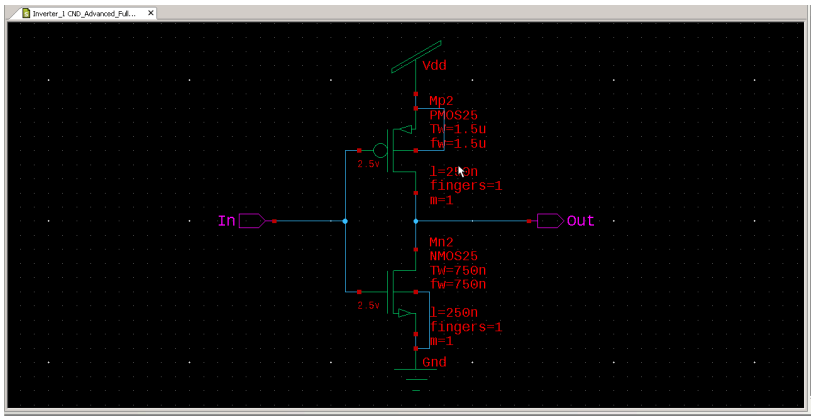


4-Top Level

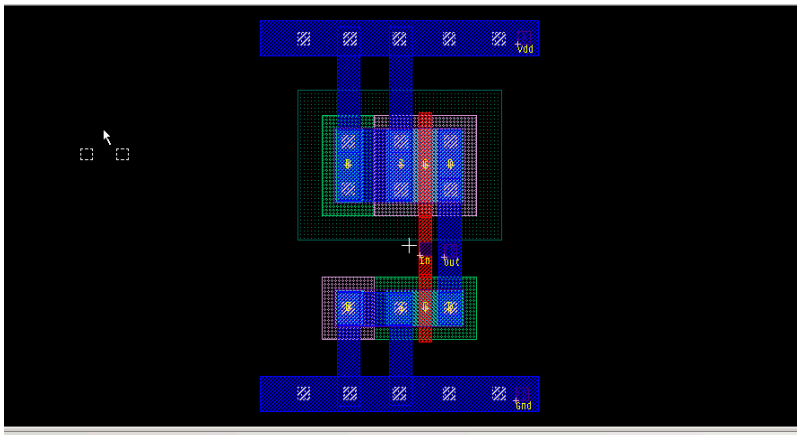




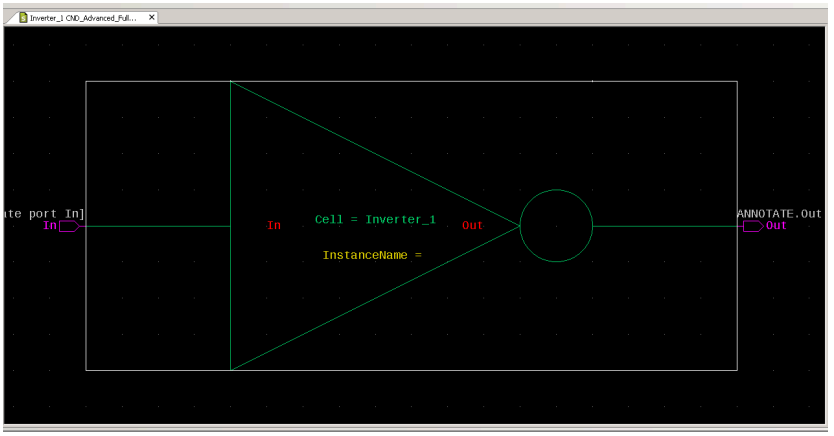
7. schematic and cell designs for each block
    - Inverter
        1. Schematic



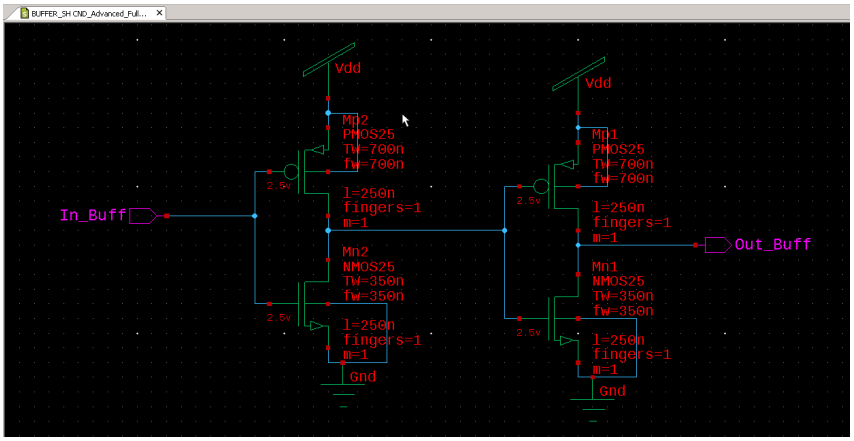
        2. Layout



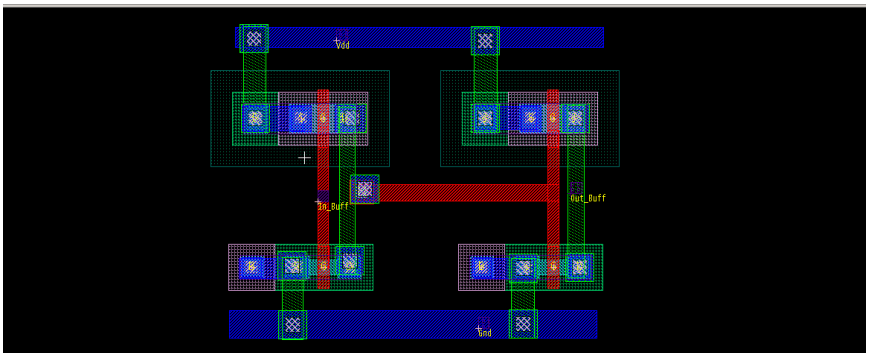
        3. Symbol



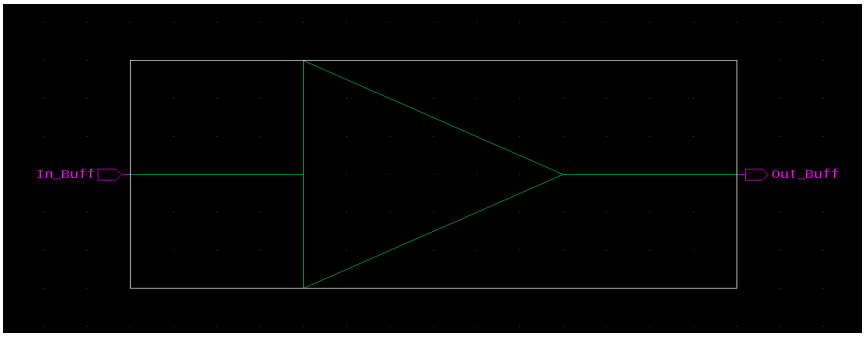
    - Buffer
        1. Schematic



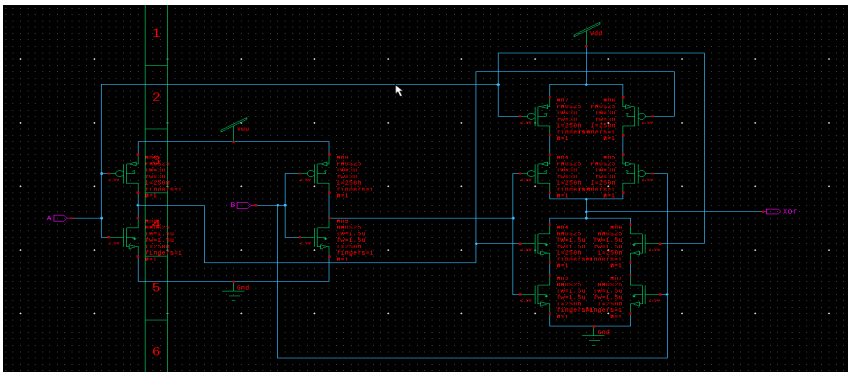
2. Layout



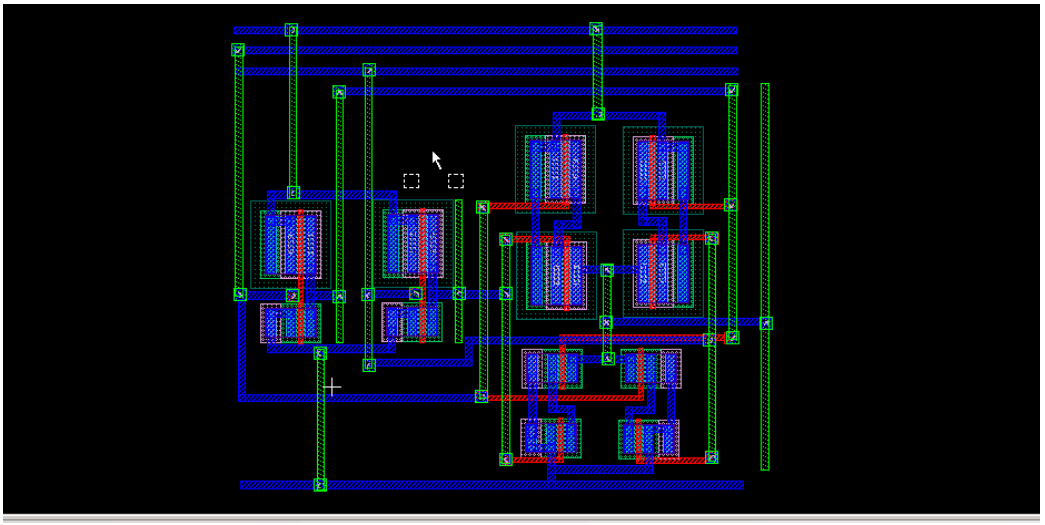
3. Symbol



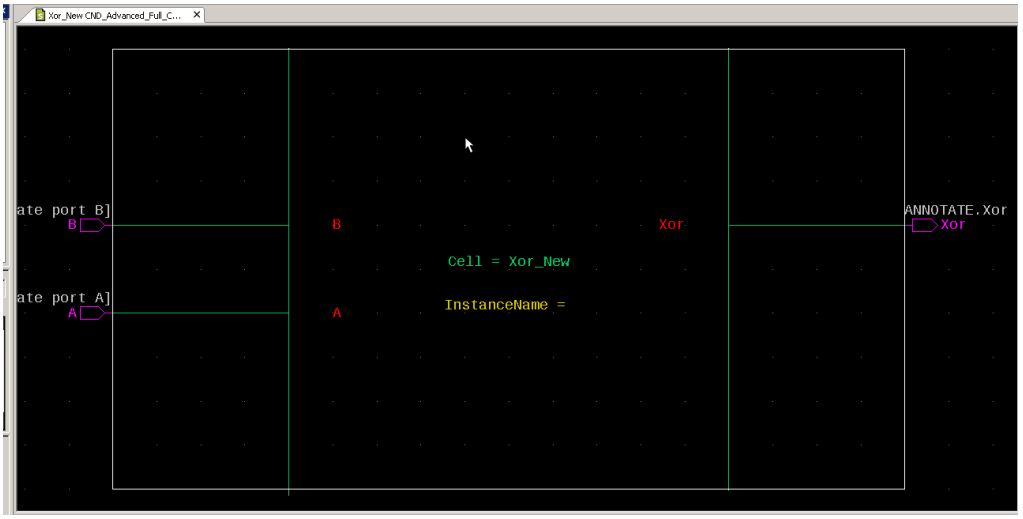
- XOR
    1. Schematic



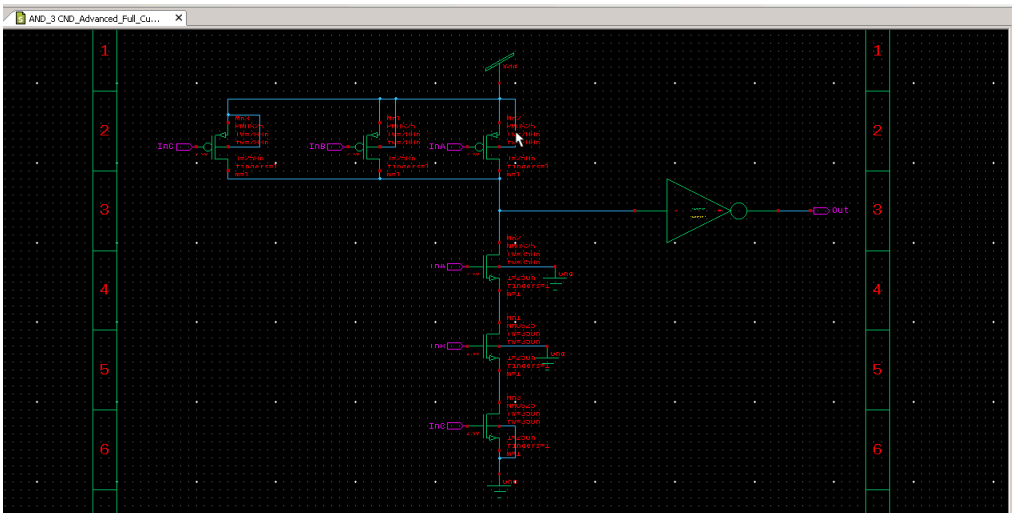
2. Layout



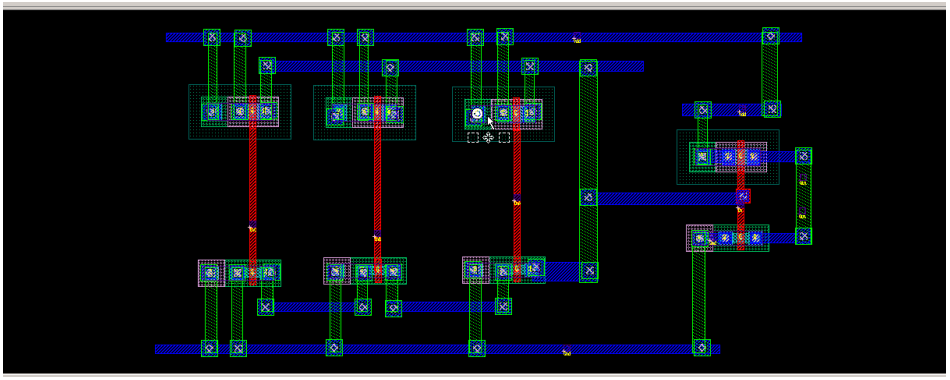
3. Symbol



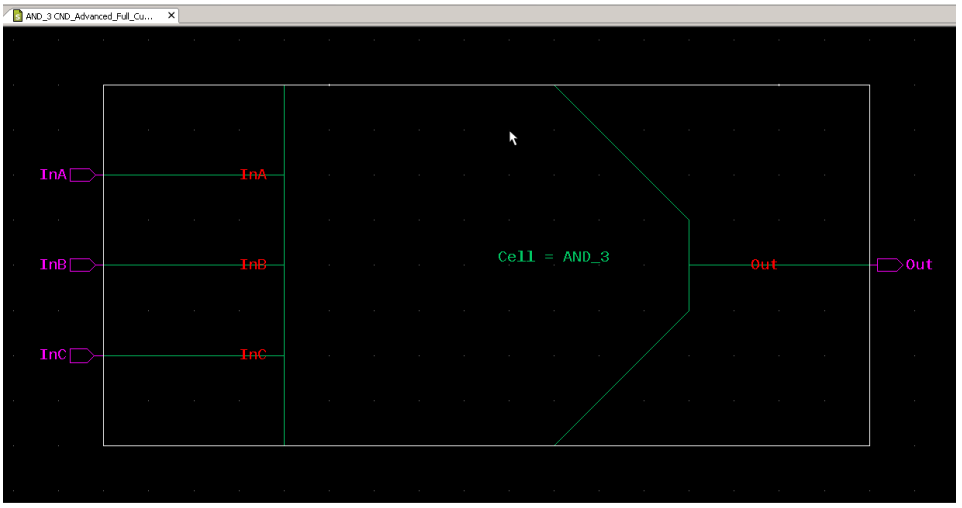
- AND
    1. Schematic



2. Layout

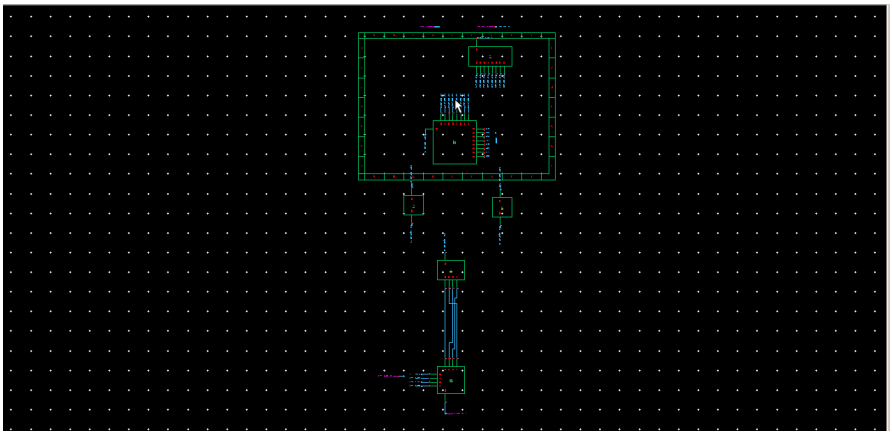


3. Symbol



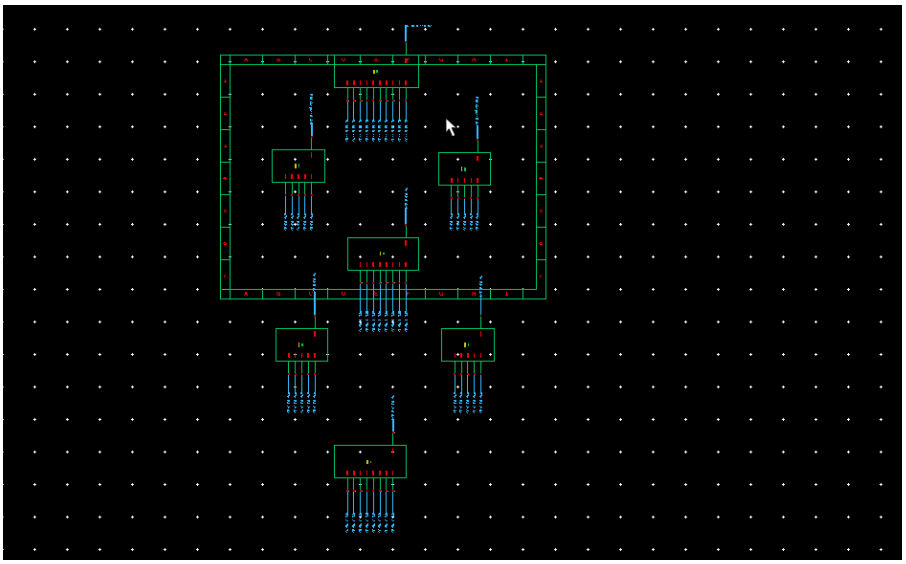
- IP
- IP-1
- SW
- P10
- LS1
- LS2
- P8

8. Final simulation results (individual blocks and integrated system)
   - FK Block
     1. Schematic

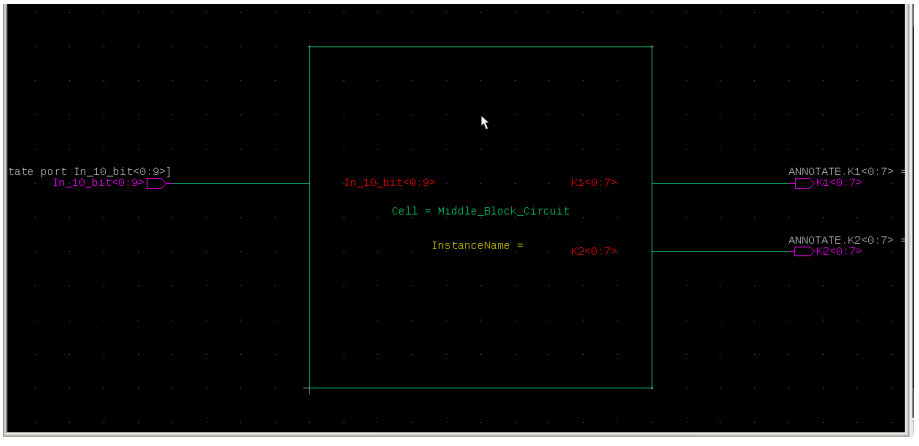


     2. Symbol

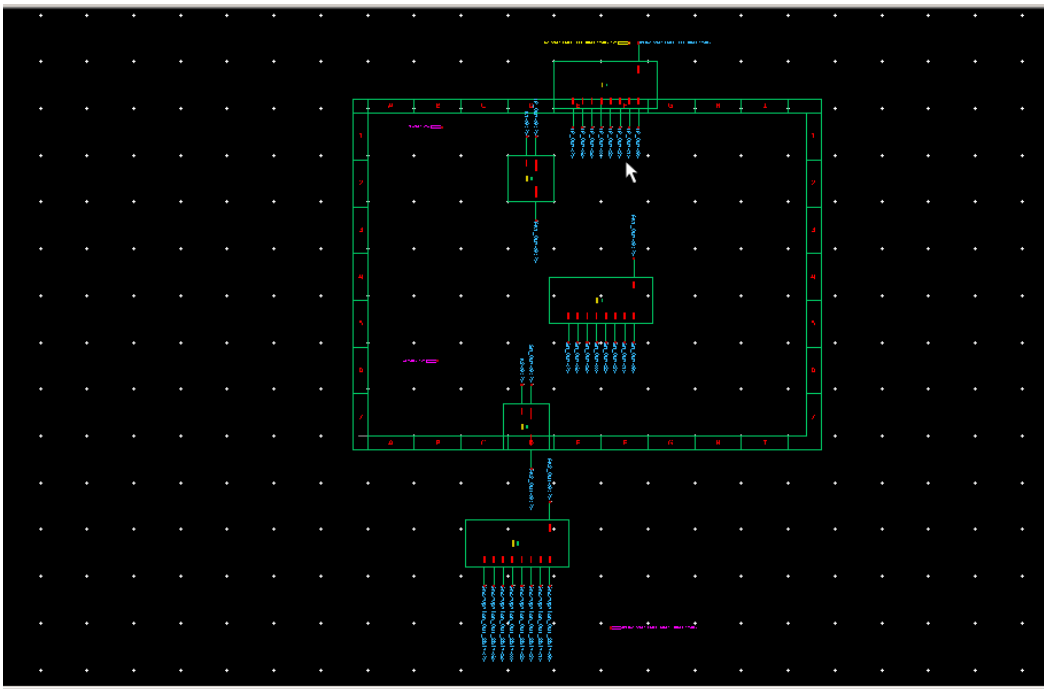
   - Key Generation Block
     1. Schematic



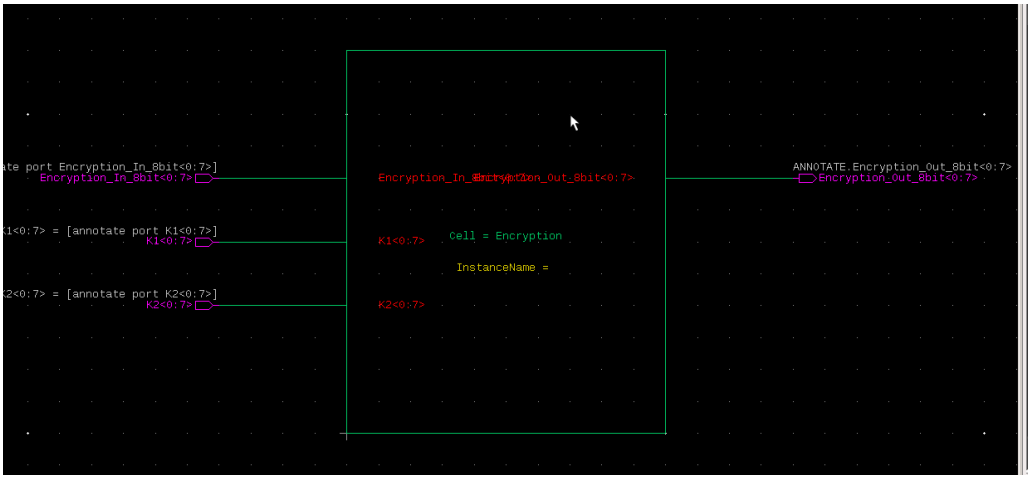
     2. Symbol



- Encryption Block
    1. Schematic



    2. Symbol



- Final Block
    1. Schematic

    2. Symbol
    3. TB
    4. Wave

9.Completed the layout of all blocks and started routing and placement

- FK Block
    1. Layout
    2. DRC
    3. LVS
- Key Generation Block
    1. Layout
    2. DRC
    3. LVS
- Encryption Block
    1. Layout
    2. DRC
    3. LVS
- Final Block
    1. Layout
    2. DRC
    3. LVS

10. Conclusion

Throughout this project, we embarked on a comprehensive exploration and optimization of the Data Encryption Standard (DES) algorithm. Our objective was to develop a fully customized implementation of DES, addressing its vulnerabilities and enhancing its security.
Design Process:

We initiated the project with meticulous system specification, delineating functional requirements and architectural constraints. This served as the foundation for our structured design process.
Employing a hierarchical structure of subsystems allowed for incremental development and testing, ensuring the creation of robust working components.
Throughout the design endeavor, paramount importance was placed on the application of design methodologies discussed in the course. Concepts such as abstraction, modularity, and design reuse guided our architectural decisions and implementation strategies.

Our custom DES implementation included advanced techniques to address vulnerabilities found in the standard DES algorithm.

We optimized the algorithm by incorporating enhanced permutation and substitution operations, as well as introducing stronger key generation mechanisms.

Rigorous verification procedures were systematically applied at each stage of development, guaranteeing the integrity and functionality of our custom DES implementation.

11. References

https://www.geeksforgeeks.org/simplified-data-encryption-standard-set-2/
https://sandilands.info/crypto/DataEncryptionStandard.html#x16-780008.2