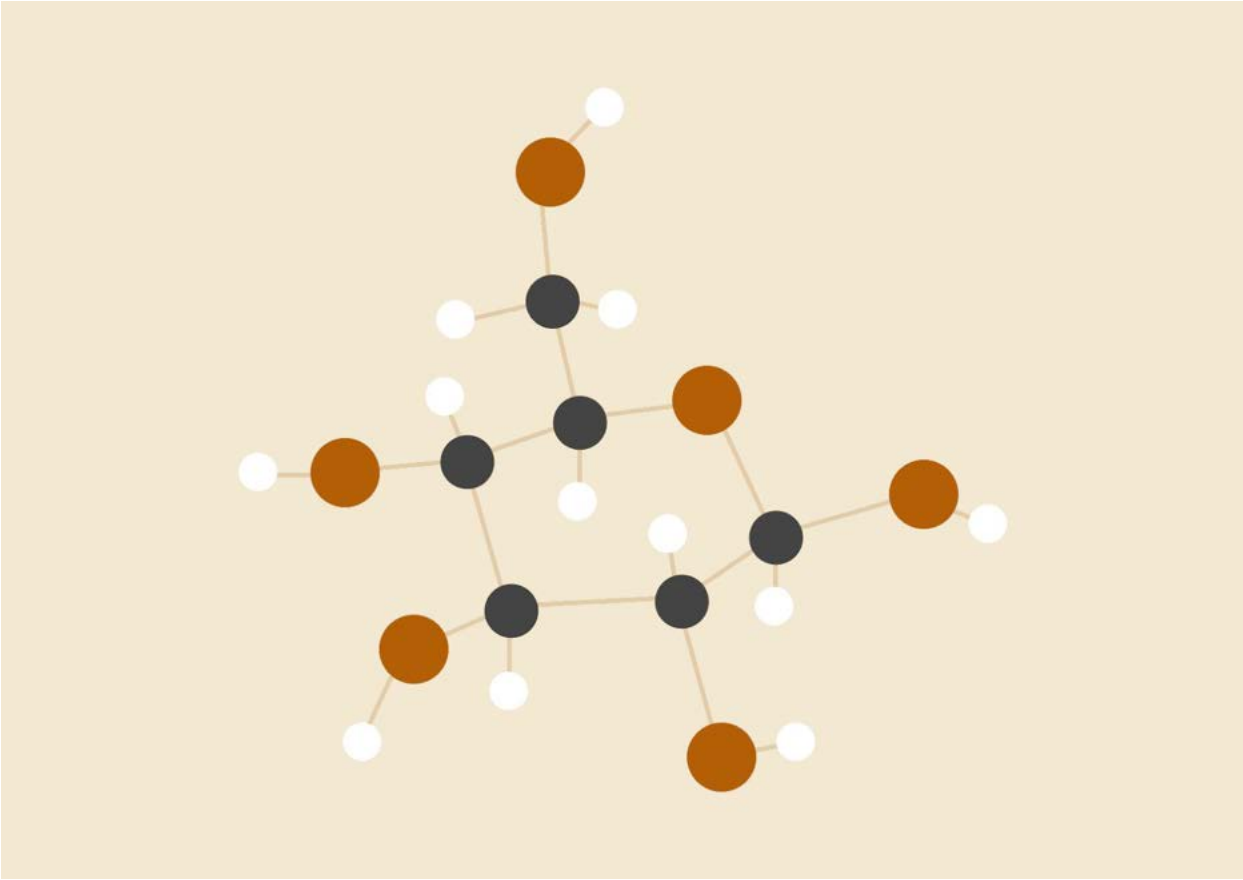
VLSI Project - Report

D.E.S ENCRYPTION



Henry Dacres | Erik Vickerd | Matt Smith

DUE 12.5.2019

CSE 493

Table of Contents

**Note: Anytime the word Views is used above, it refers to the schematic view, layout view, extracted view, simulation of the schematic, simulation of the extracted, and LVS check. **

Introduction:

The Data Encryption Standard (DES) is our VLSI project for CSE 493. We chose to do DES encryption due to our interest in encryption. In doing this project, we all have a better understanding of how this algorithm works after doing this project. DES encryption was developed by IBM in 1975 and is a symmetric encryption algorithm. This simply means that it uses the same key for encryption and decryption. Typical DES encryption takes in a 64-bit input, a 64-bit key, contains 16 rounds and outputs a ciphered 64-bit text. The 64-bit key is reduced to 56-bits due to every 8th bit being used for parity. Due to limited time and constraints, we downscaled from 64-bits to 16-bits. Our DES system inputs 16-bits, a 12-bit key, contains 4 rounds and outputs a 16-bit ciphered text. The encryption generation process is detailed in the next section.

Encryption Generation Process

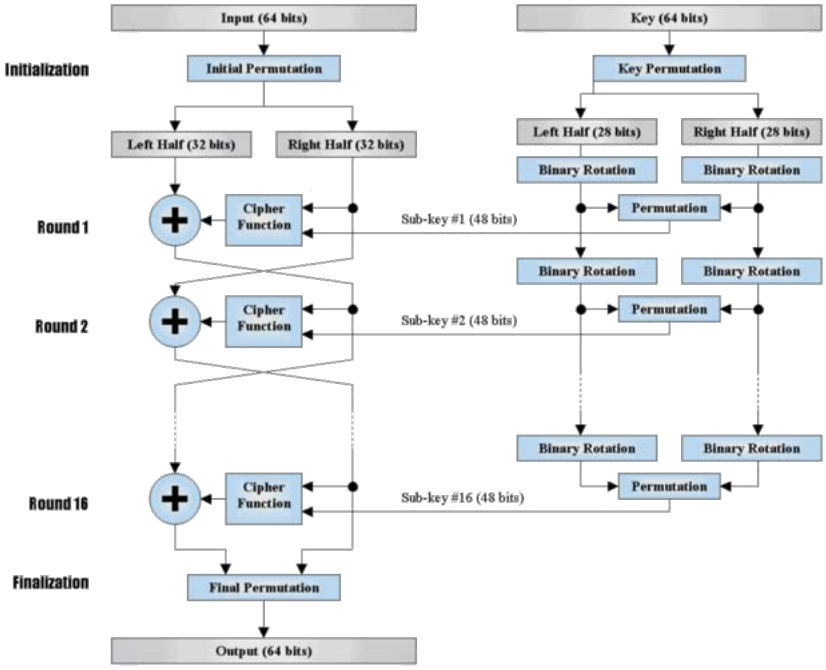


Figure 1: Process of Encryption Generation and Subkey Generation [1]

The key is selected at random. After that, the key goes through Permutation Choice 2 and is split into two halves. Each half goes through a binary rotation depending on the round number. Then, the two rotated halves go into a Permutation. The result of this Permutation is a subkey. To generate an encrypted word, our input goes through an Initial Permutation function listed below. After that, the input is split into two halves, the left half and the right half. The right half goes through a Cipher function with the subkey for that specific round. The Cipher function goes through the following order. The right half is put into an Expansion function. After that, the expanded right half is then XORed with the sub key. Then, the output from the XOR gates goes through the S-box functions. Lastly, this goes through a Permutation function. After the right half goes through the Cipher function, it is added with the left half. The last carryout is dropped. The new right half is the sum that is produced from this function. The new left half is the old right half. This process repeats for 4 rounds. After the 4th round, the left half and the right half are put into a Final Permutation. The output will be the encrypted word.

Subkey Generation:

- **Permutation Choice 1**

Permutation Choice 1 does a port map of the input key to make it more difficult to see where the key comes from. Listed below is that specific port map.

Permutation Choice 1			
13	9	5	1
14	10	15	11
7	3	6	2

Table 1: Permutation Choice 1

- Circular Shift Left Functions

Each circular shift left function occurs when a new subkey needs to be generated. Depending on the round number, either a left shift by 1 or 2 will occur. Rounds 1, 2, 9, and 16 are the left shift by 1. Every other shift left will be by 2. Listed below are the port maps for circular shifting right and left.

Circular Shift Left by 1					
6	1	2	3	4	5

Table 2: Circular Shift Left by 1

Circular Shift Left by 2					
6	5	1	2	3	4

Table 3: Circular Shift Left by 2

- Permutation Choice 2

After the binary rotations on the left and right halves, each half is port mapped into the permutation choice 2 function. The permutation choice 2 function is the function that creates the subkey's which are then XORed with the right half of the input. Listed below is the Permutation Choice 2 function.

Permutation Choice 2			
4	3	6	5
2	1	8	12
7	9	10	11

Table 4: Permutation Choice 2

Encryption:

- **Initial Permutation**

The initial permutation is used to protect the “word” entered into the system. The first thing done by the encryption system is the initial permutation function. Listed below is the port map for the initial permutation.

Initial Permutation			
14	10	6	2
16	12	8	4
13	9	5	1
15	11	7	3

Table 5: Initial Permutation

- Cipher Function

Before entering the cipher function, the input is split into two halves, the right and left halves. The right half enters the cipher function while the left half is added to the result of the cipher function. The new right half is the sum of the left half and the cipher function result from the right half. The new left half is the old right half (before modification through the cipher function). Listed below are the steps taken to get through the cipher function.

  - **Expansion Function**

When the right half enters the cipher function, it enters the expansion function. The expansion function increases the size of the right half of the input. This is needed due to the size of the sub-key function. Listed below is the port map for the expansion function

Expansion Function					
8	1	2	3	4	5
4	5	6	7	8	1

Table 6: Expansion Function

The shaded areas are where the expansion took place. Inputs 1,4,5,and 8 are all repeated to gain a larger input.

  - **XOR**

The two inputs to the XOR gate are the expanded input and the generated subkey. The two inputs are XORed and continue to the S-box functions.

- **S-boxes**

For 16-bit encryption, we have four S-box functions. The XORed function splits into 4 parts, hence why there are four S-boxes. The functionality below better describes the functionality of the S-boxes.

A	B	C	S-box 1		S-box 2		S-box 3		S-box 4	
0	0	0	1	1	1	1	1	0	0	1
0	0	1	0	1	0	0	0	0	1	1
0	1	0	1	1	1	0	1	0	1	1
0	1	1	0	0	1	1	1	1	0	0
1	0	0	0	0	0	0	1	1	1	1
1	0	1	1	1	1	1	0	1	1	0
1	1	0	0	1	0	1	0	0	1	0
1	1	1	0	1	1	0	1	0	0	1

Table 7: S-box functionality

The inputs listed A,B, and C are the split inputs from the XOR function. There are four of these inputs to each S-box function. Each S-box function acts like a LUT.

- **Permutation**

The permutation function port maps the outputs from the s-boxes. Listed below is the permutation.

Permutation			
4	5	7	3
6	2	8	1

Table 8: Permutation Function

After the s-box functions, the system is restored back to 8 bits. After the permutation port map, the left half is finally added to the modified right half.

- Final Permutation

The final permutation is the last thing done in the encryption circuit. It does one last port map to enhance the security of the process. Listed below is the port map for the final permutation. After the final permutation is done, the input has been entirely encrypted.

Final Permutation			
12	4	16	8
11	3	15	7
10	2	14	6
9	1	13	5

Table 9: Final Permutation

Initial Permutation - Views:

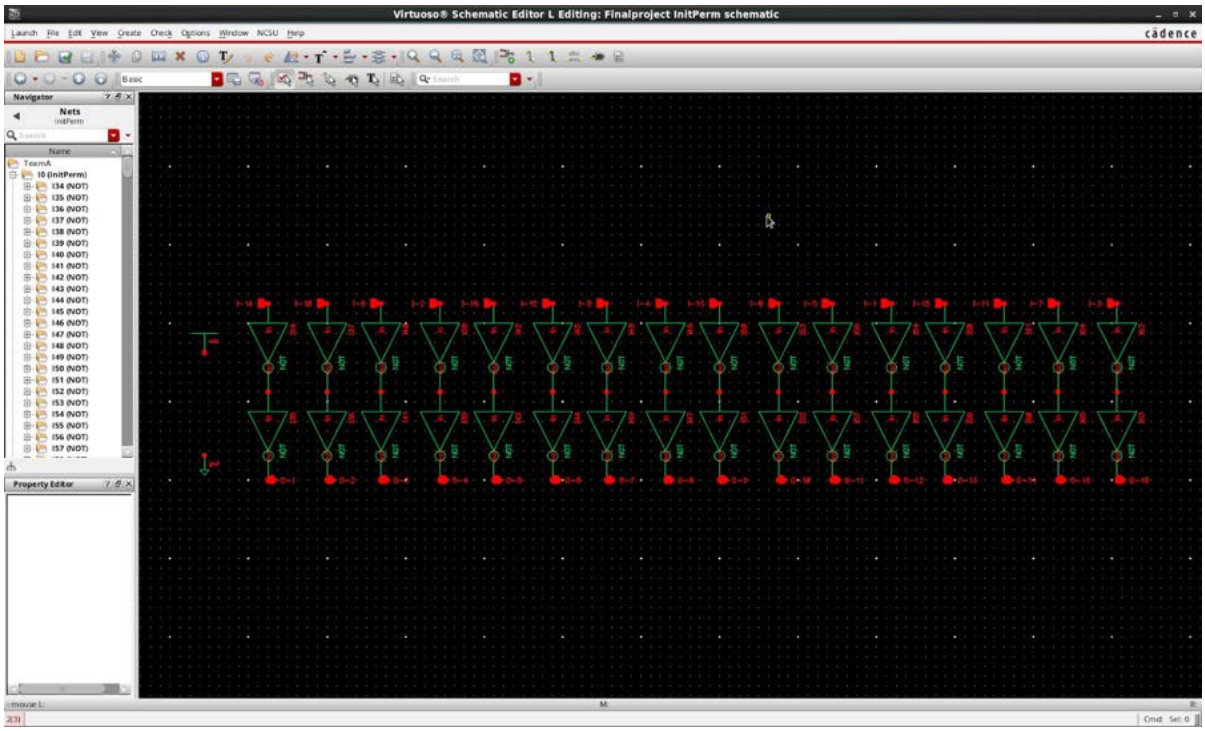


Figure 2: Schematic View of the Initial Permutation

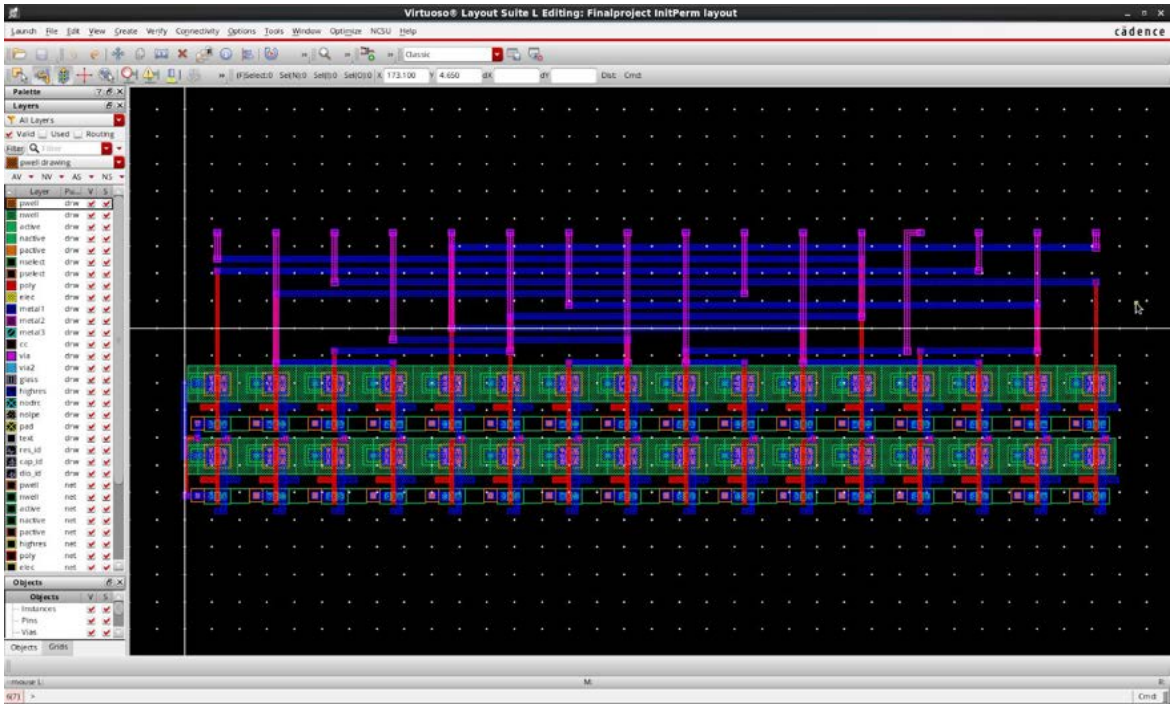


Figure 3: Layout View of the Initial Permutation

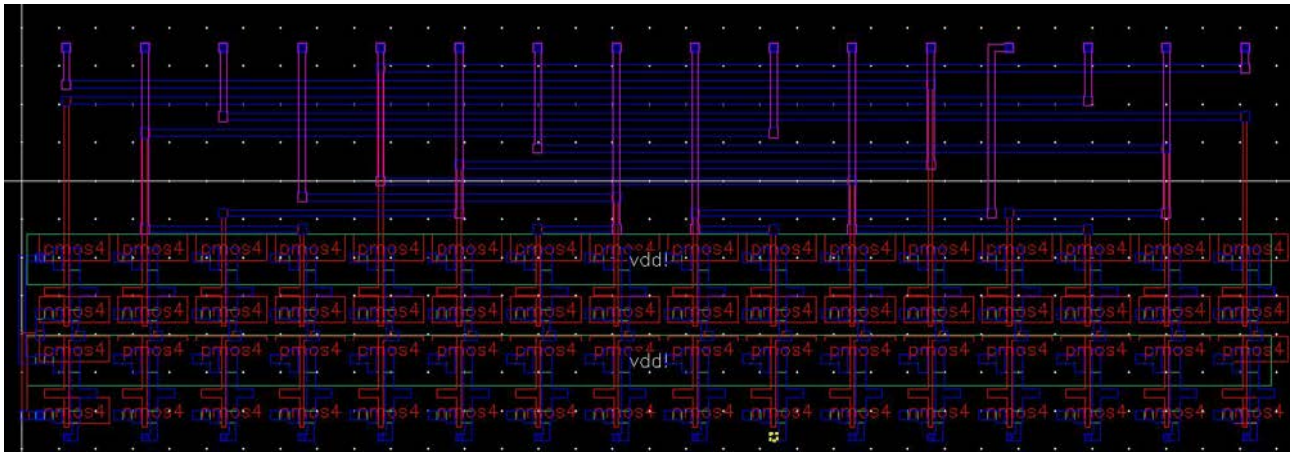


Figure 4: Extracted View of the Initial Permutation

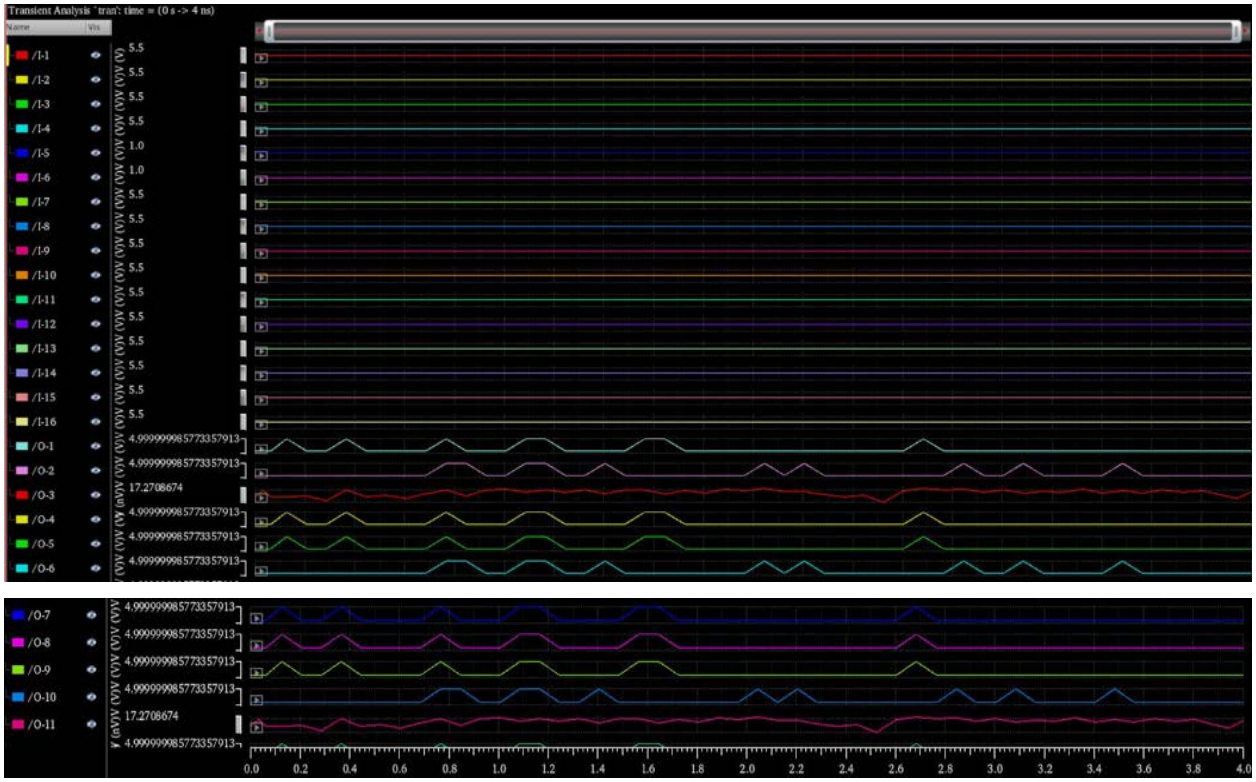


Figure 5: Schematic Simulation of the Initial Permutation

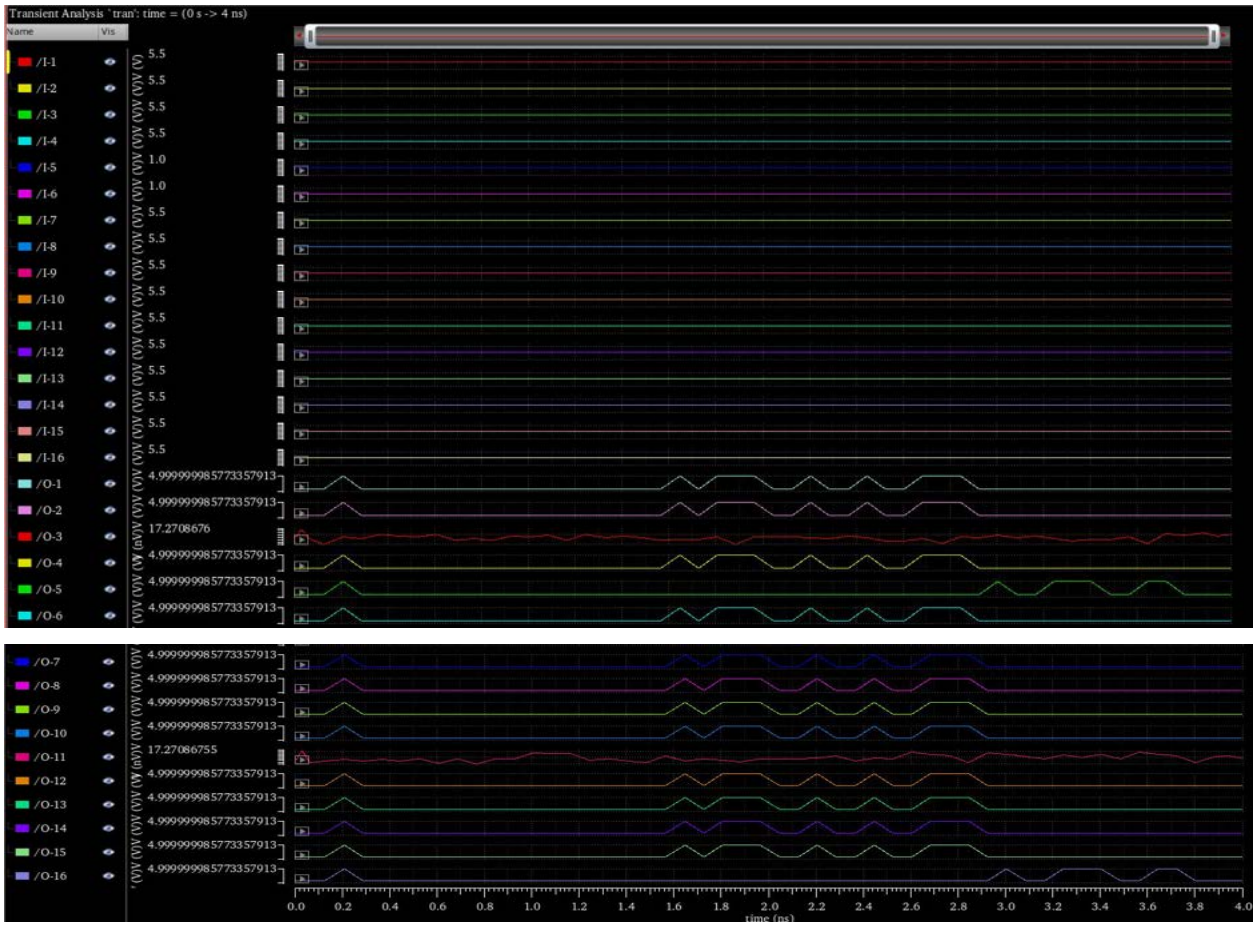


Figure 6: Extracted Simulation of the Initial Permutation

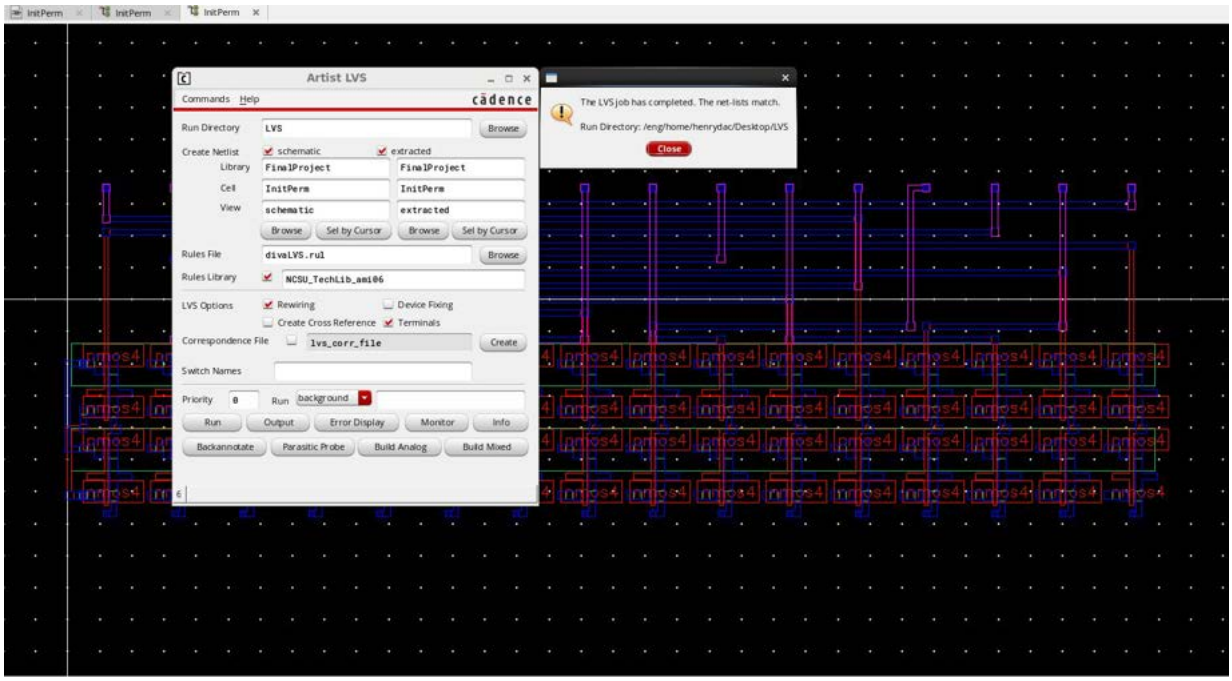


Figure 7: LVS Check for the Initial Permutation

Final Permutation - Views:

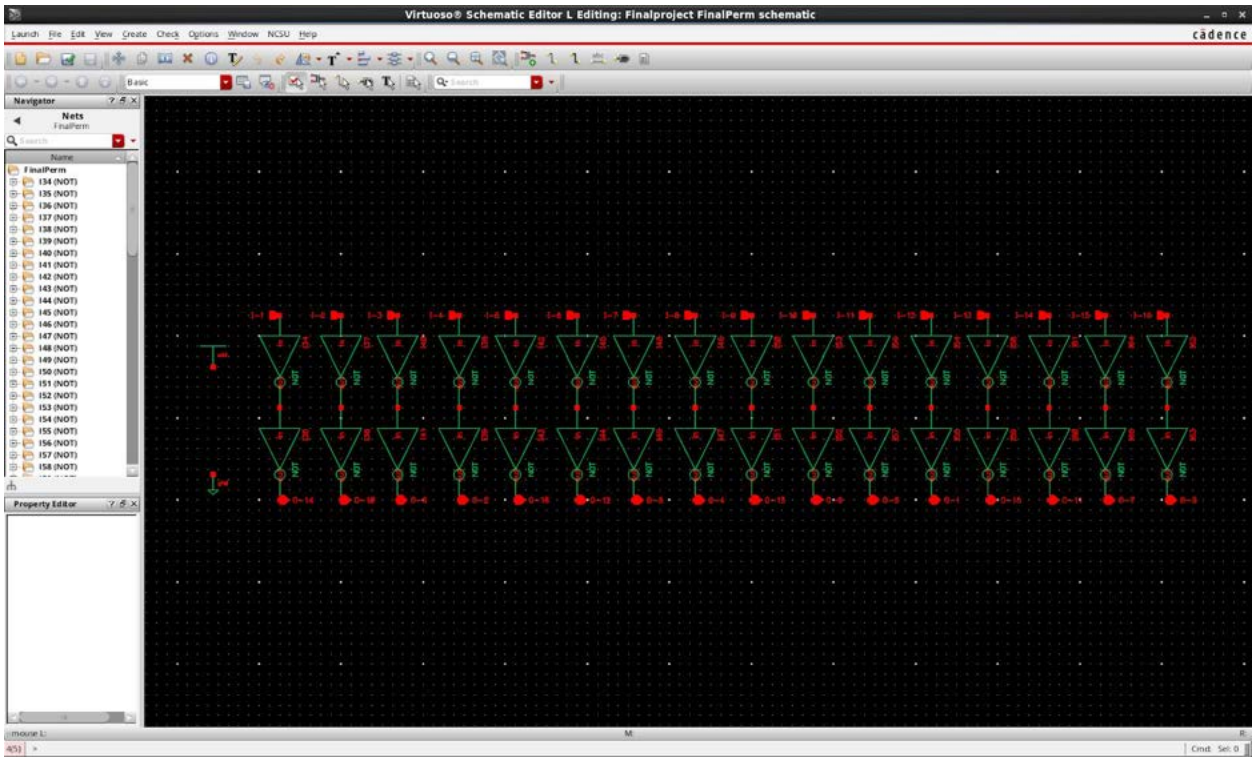


Figure 8: Schematic View of the Final Permutation

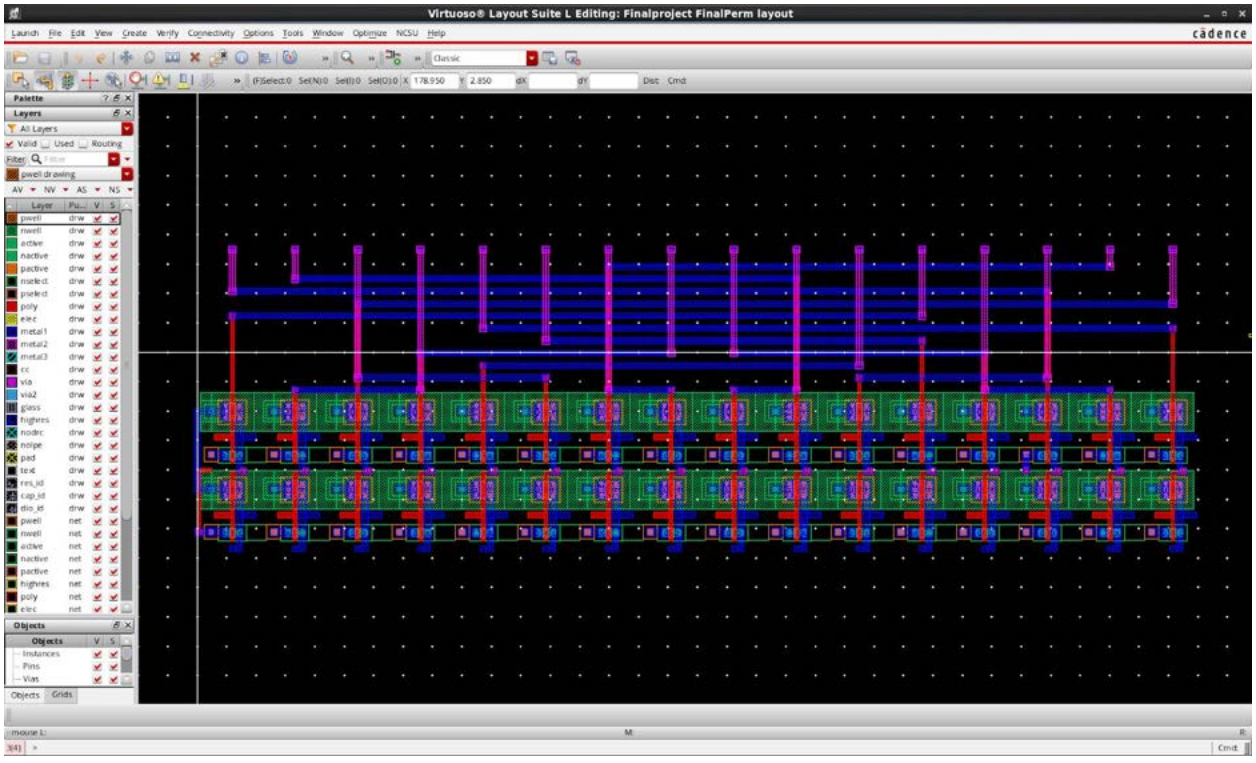


Figure 9: Layout View of the Final Permutation

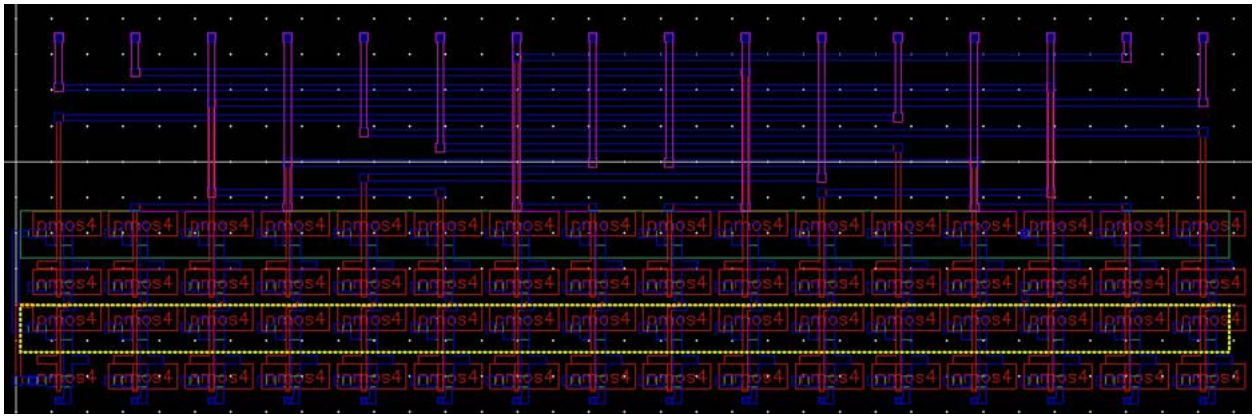


Figure 10: Extracted View of the Final Permutation

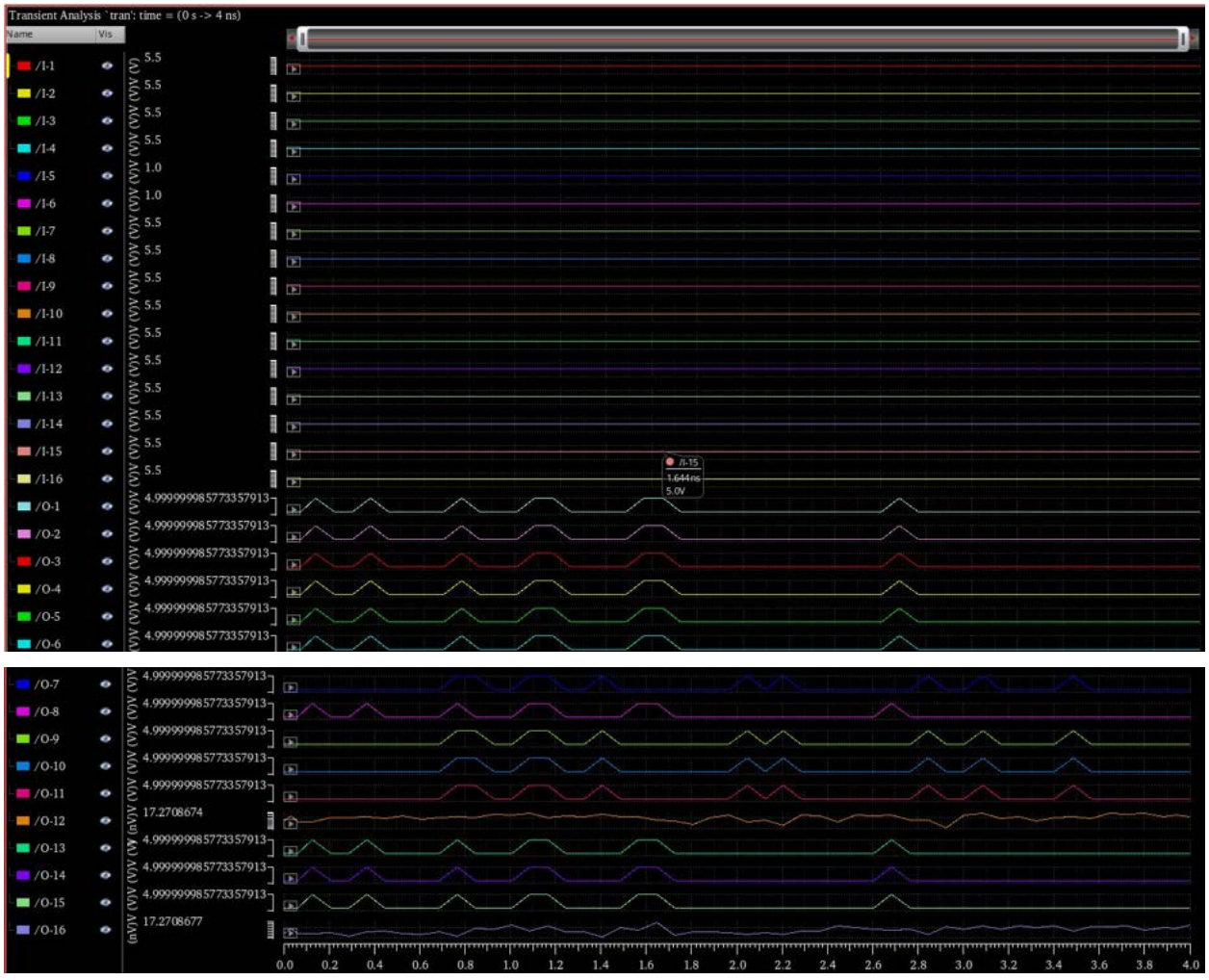


Figure 11: Schematic Simulation of the Final Permutation

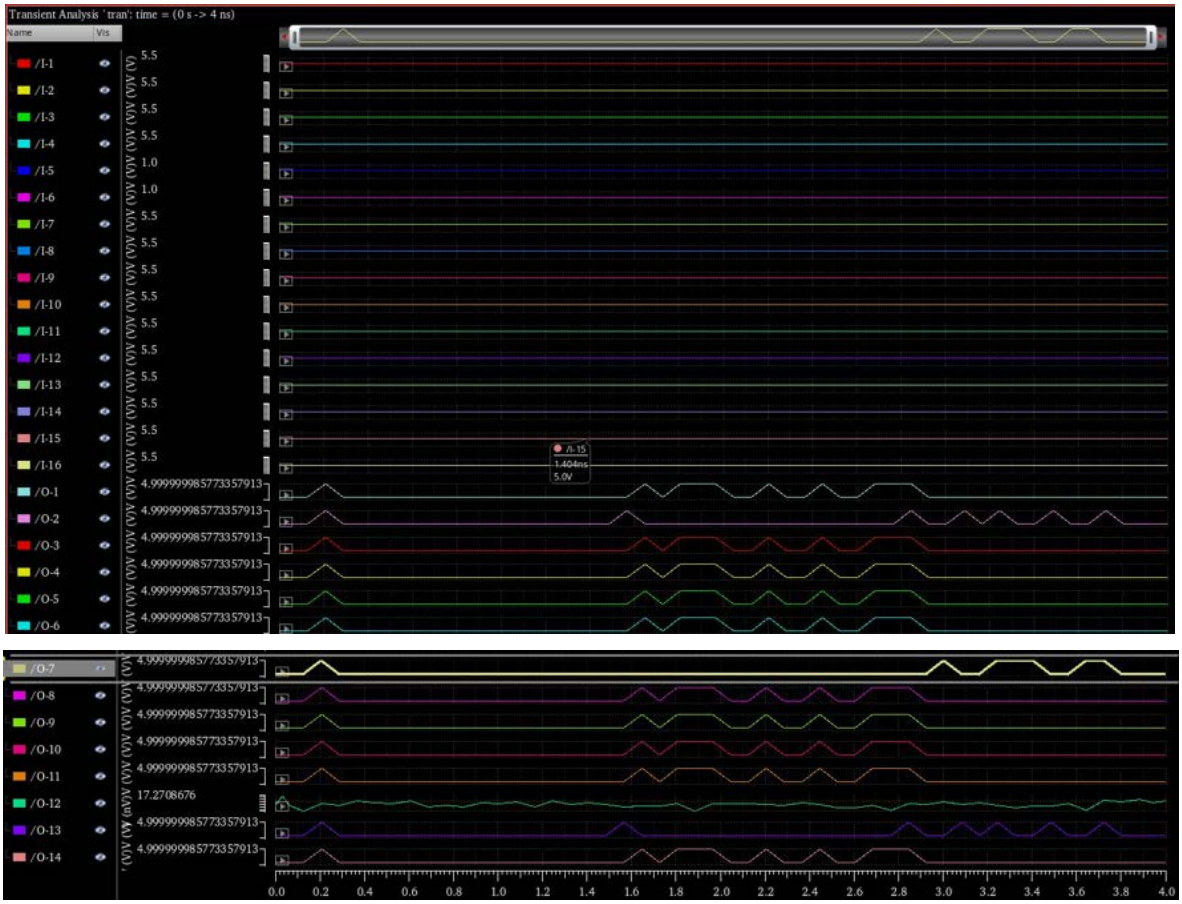


Figure 12: Extracted Simulation of the Final Permutation

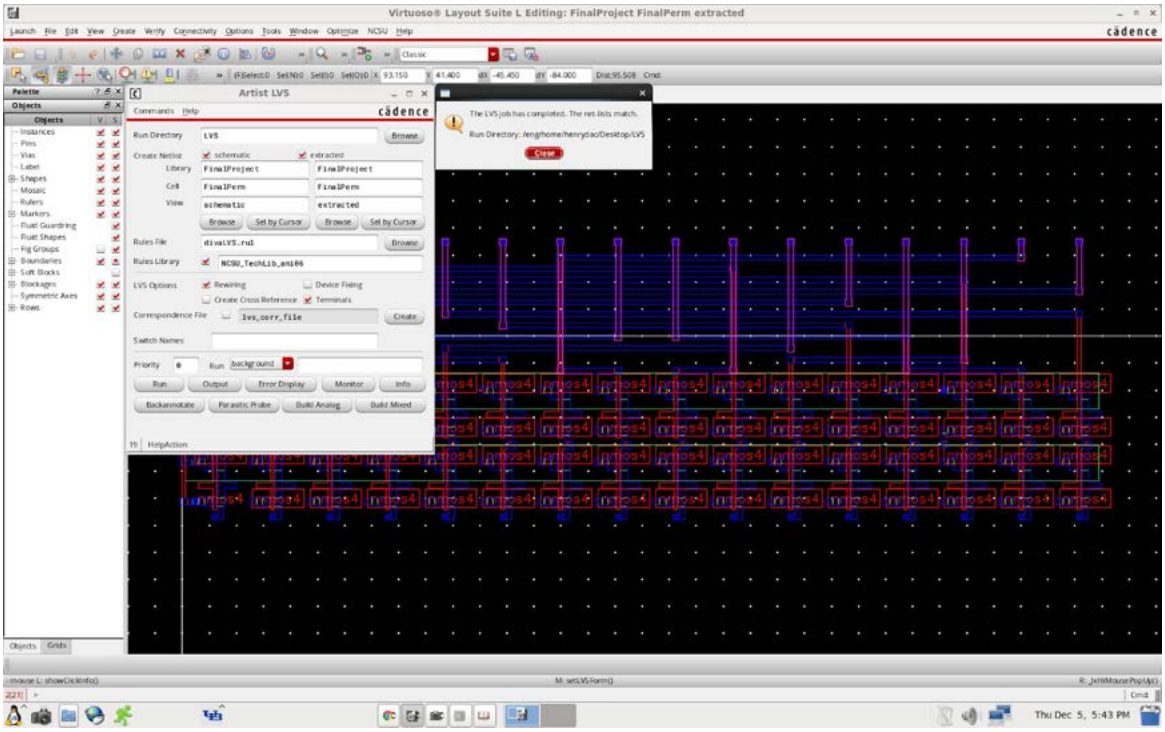


Figure 13: LVS Check for the Final Permutation

Permutation - Views:

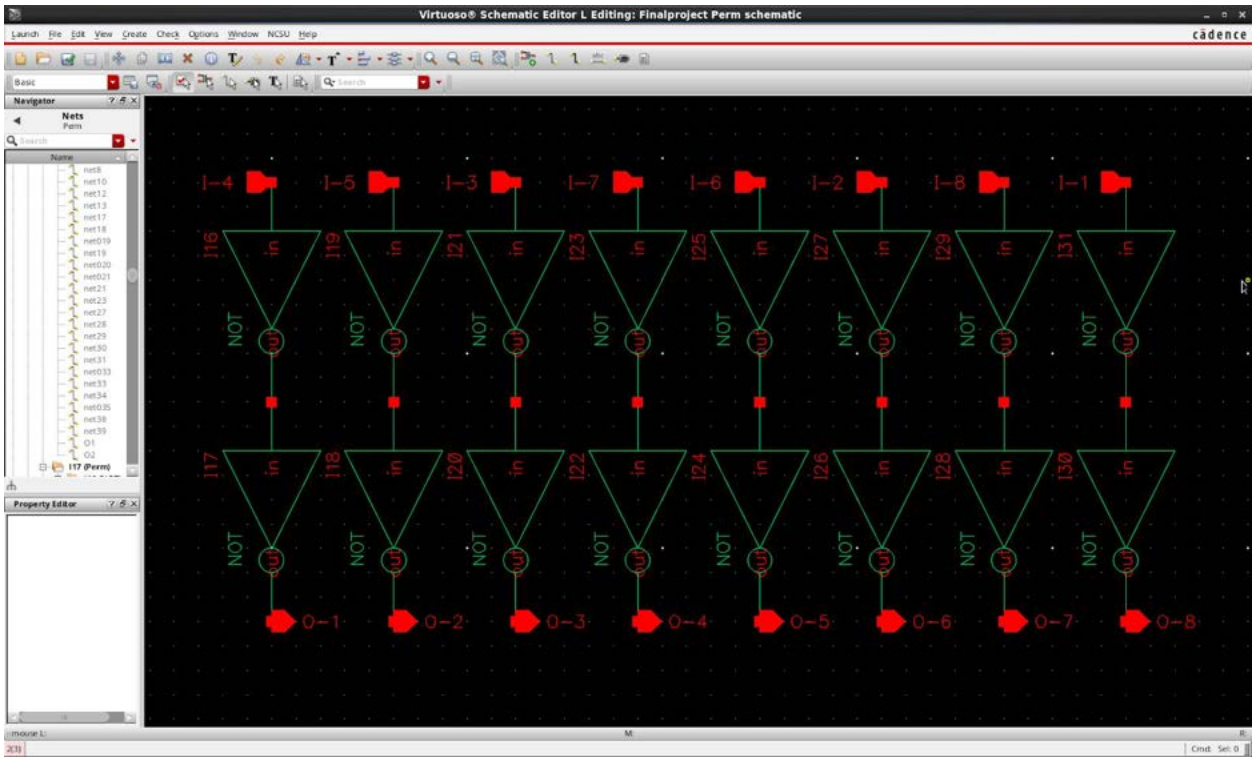


Figure 14: Schematic View of the Permutation

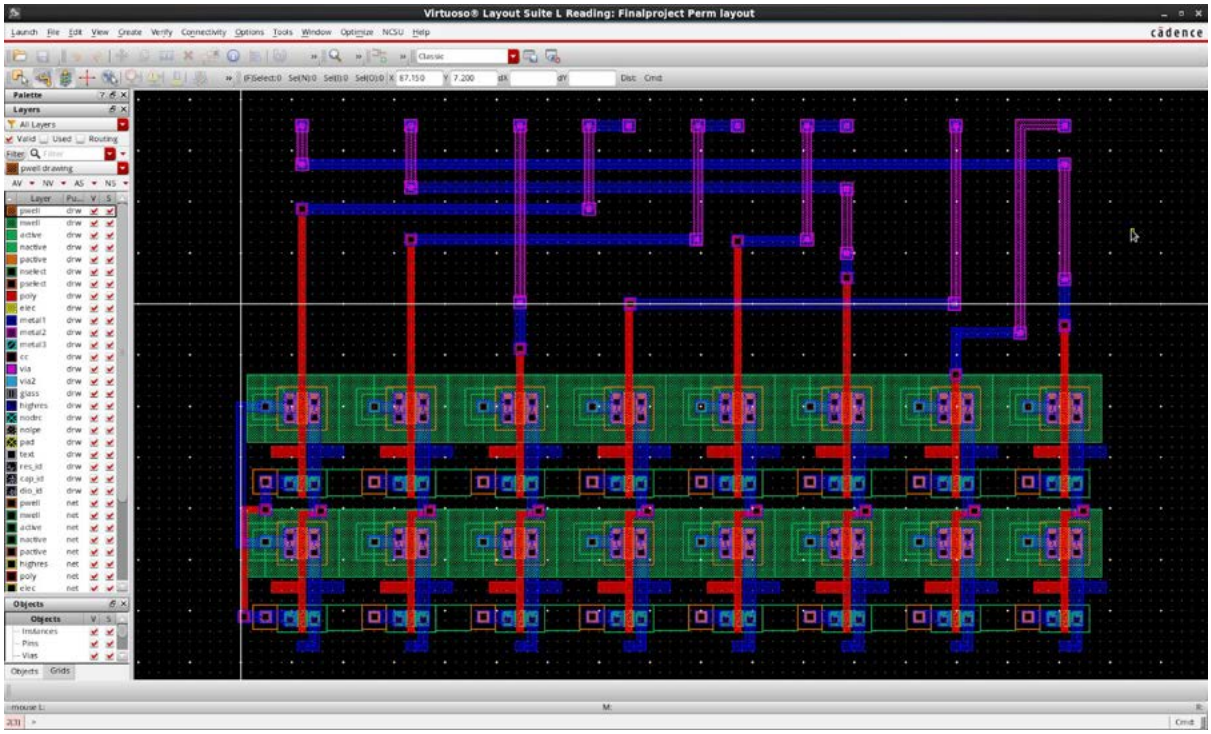


Figure 15: Layout View of the Permutation

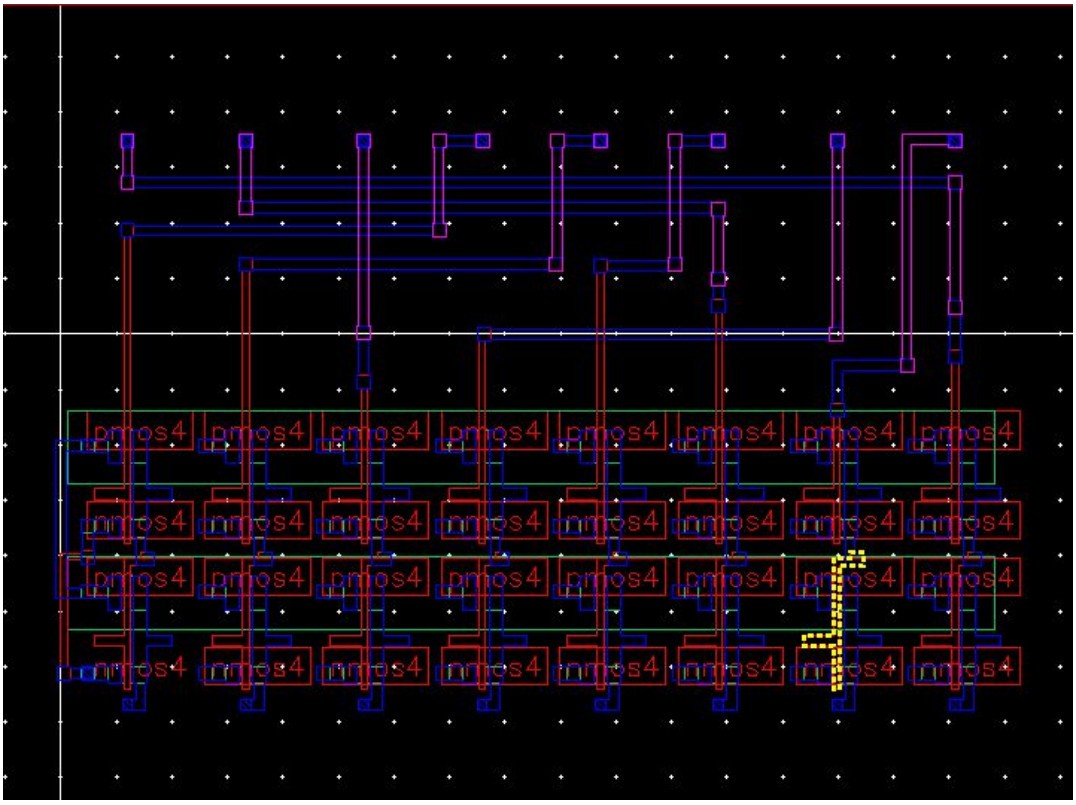


Figure 16: Extracted View of the Permutation

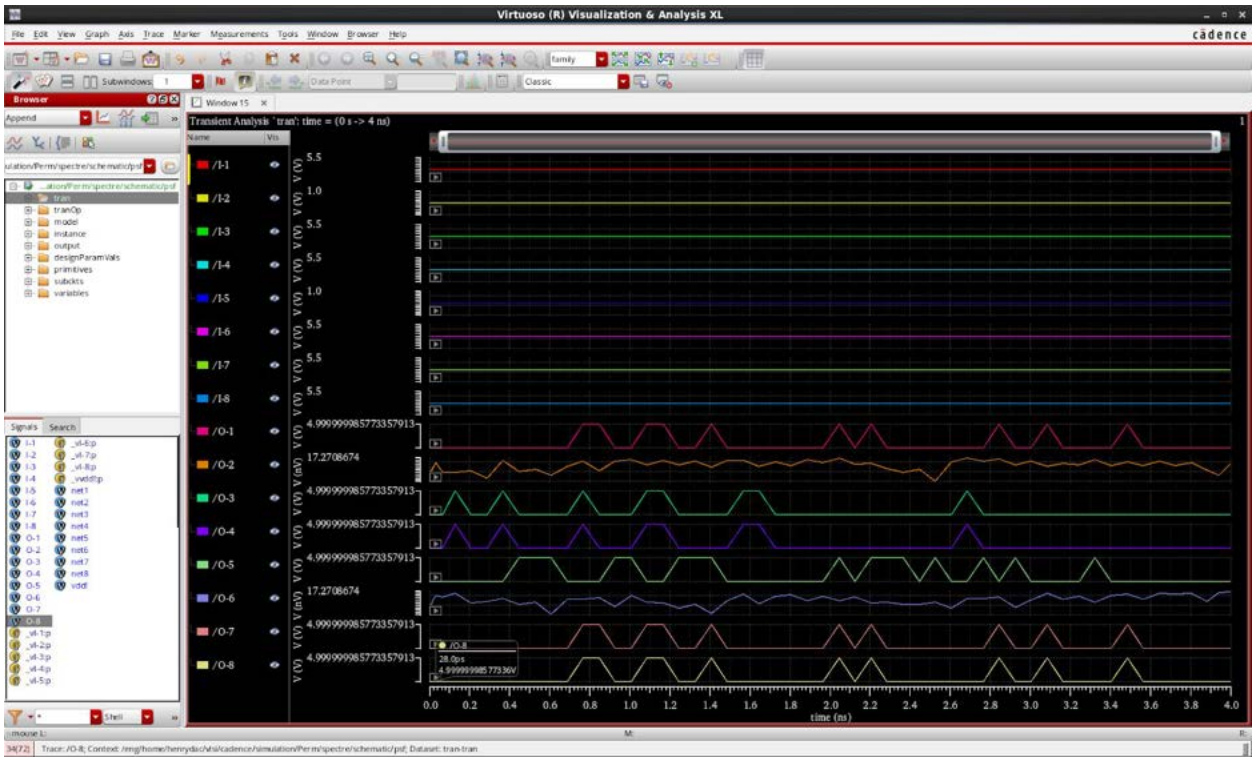


Figure 17: Schematic Simulation of the Permutation

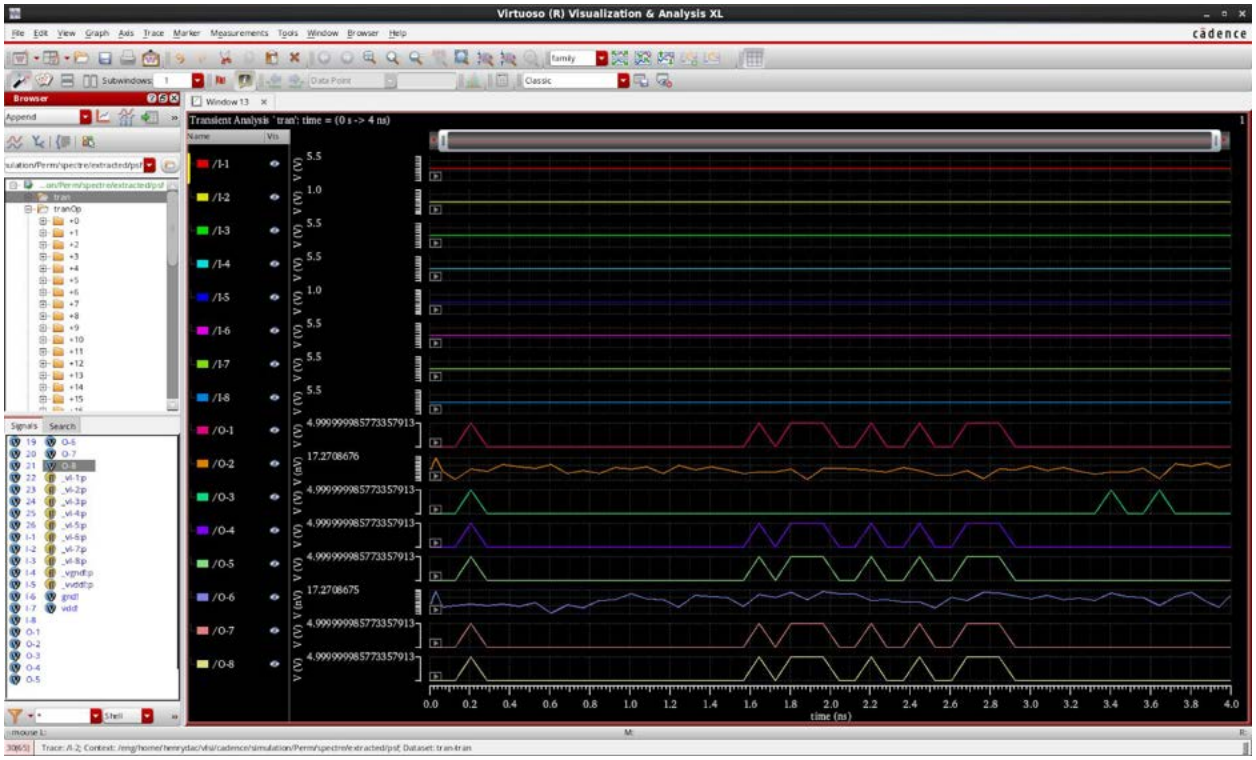


Figure 18: Extracted Simulation of the Permutation

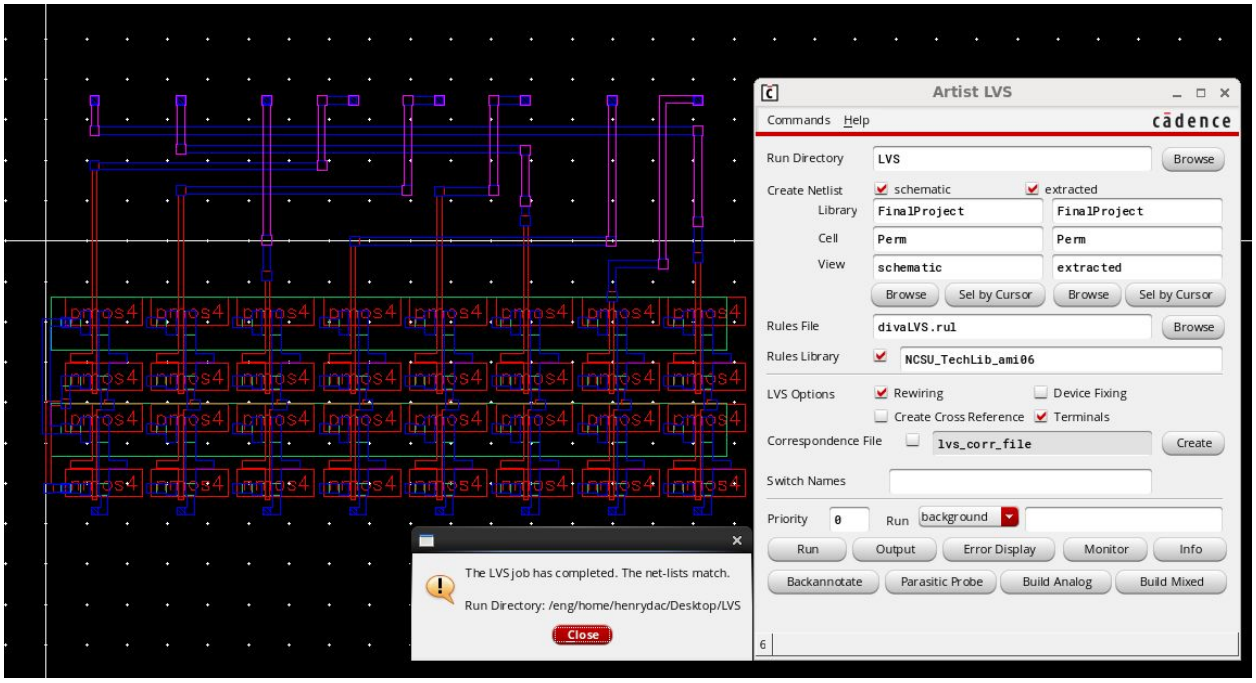


Figure 19: LVS Check for the Permutation

Permutation Choice 1 - Views:

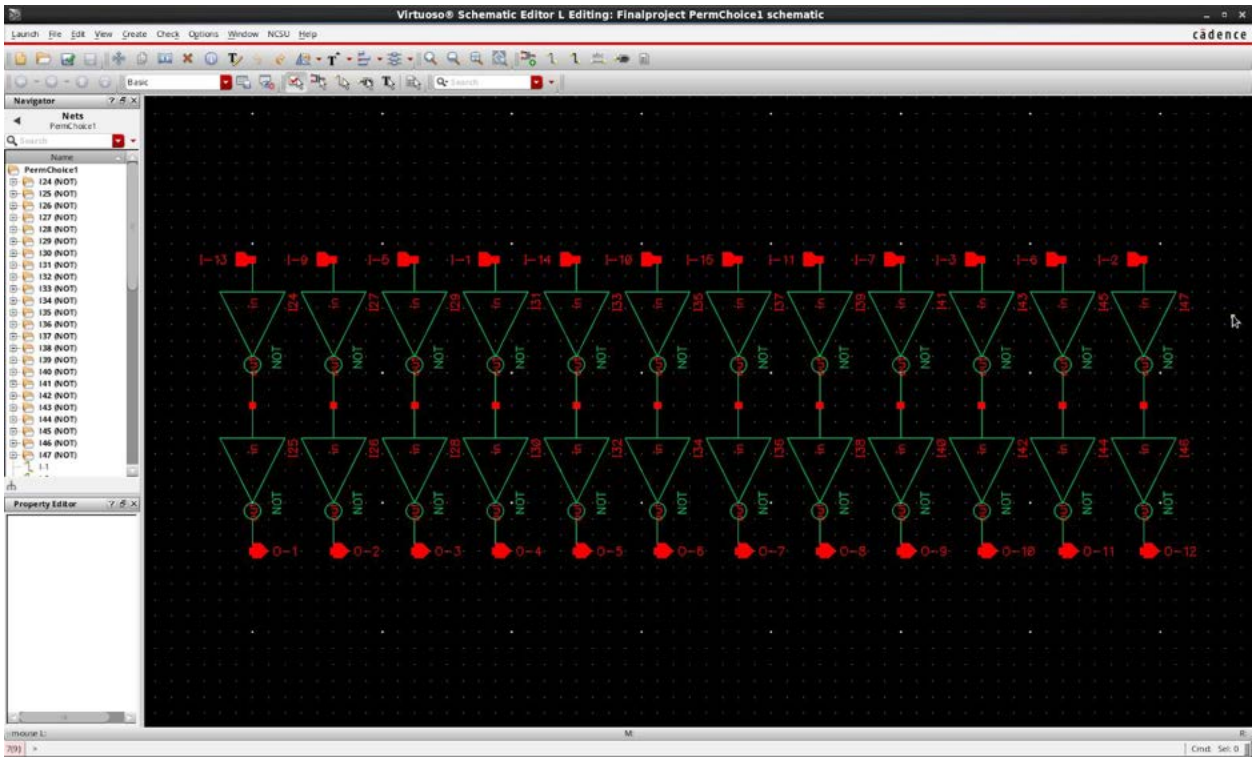


Figure 20: Schematic View of Permutation Choice 1

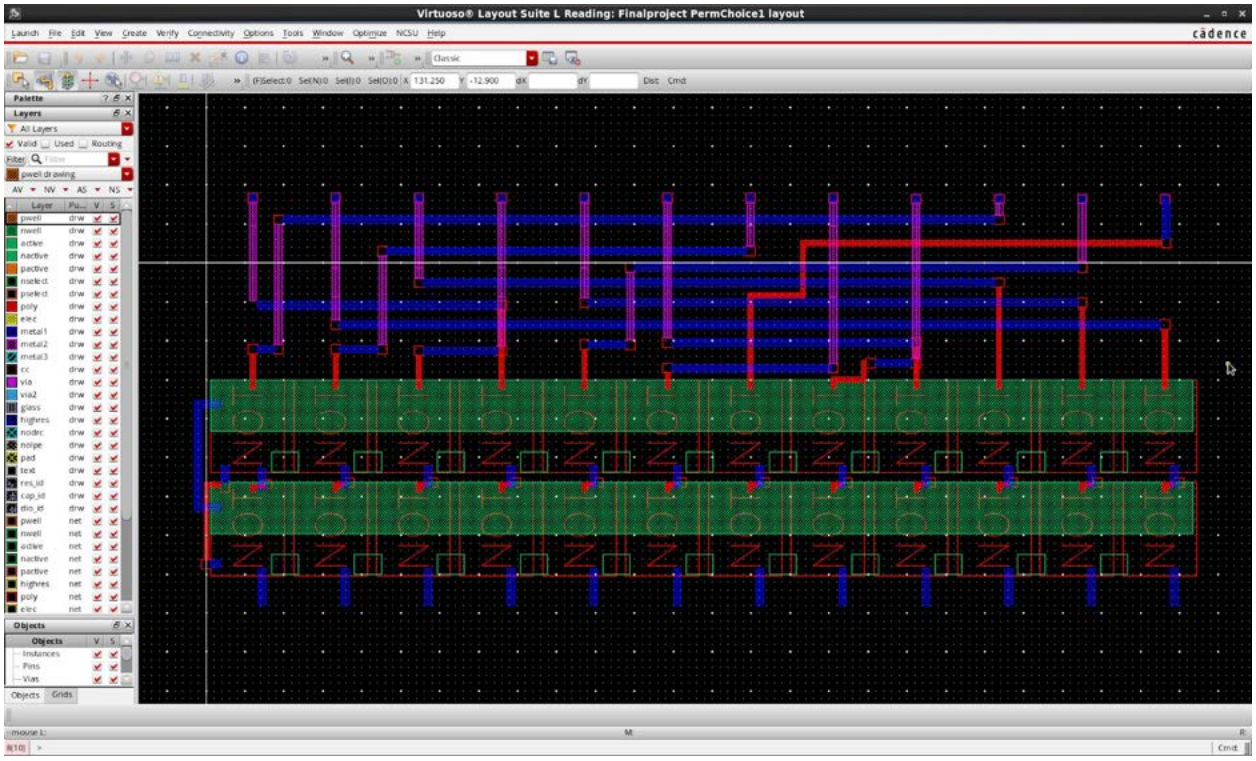


Figure 21: Layout View of Permutation Choice 1

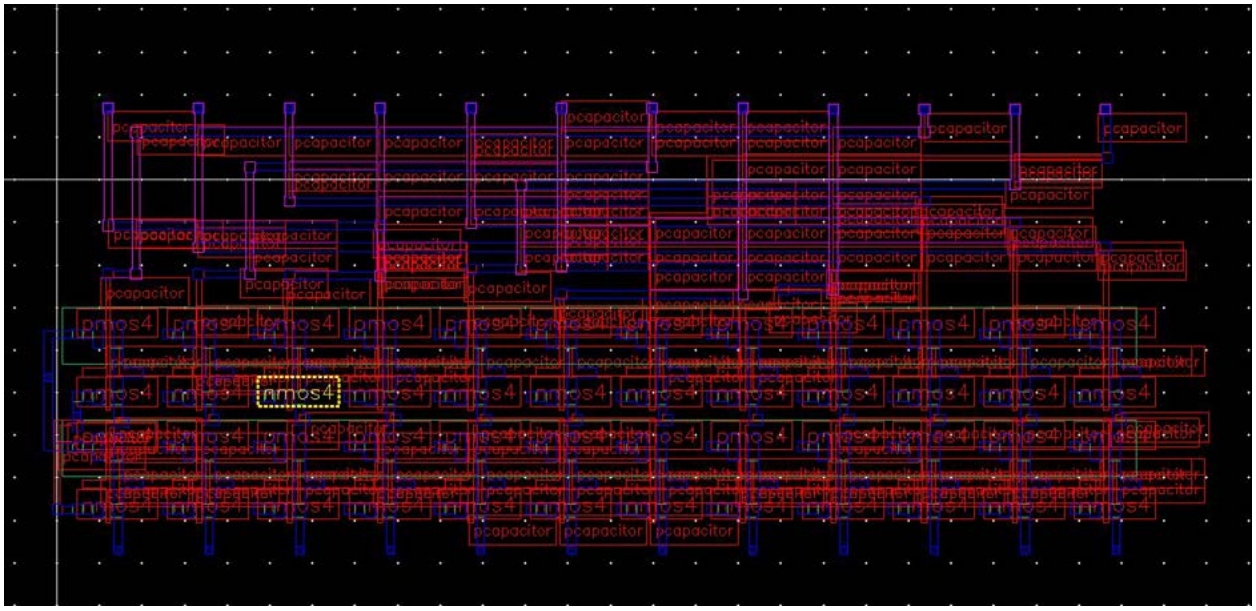


Figure 22: Extracted View of Permutation Choice 1

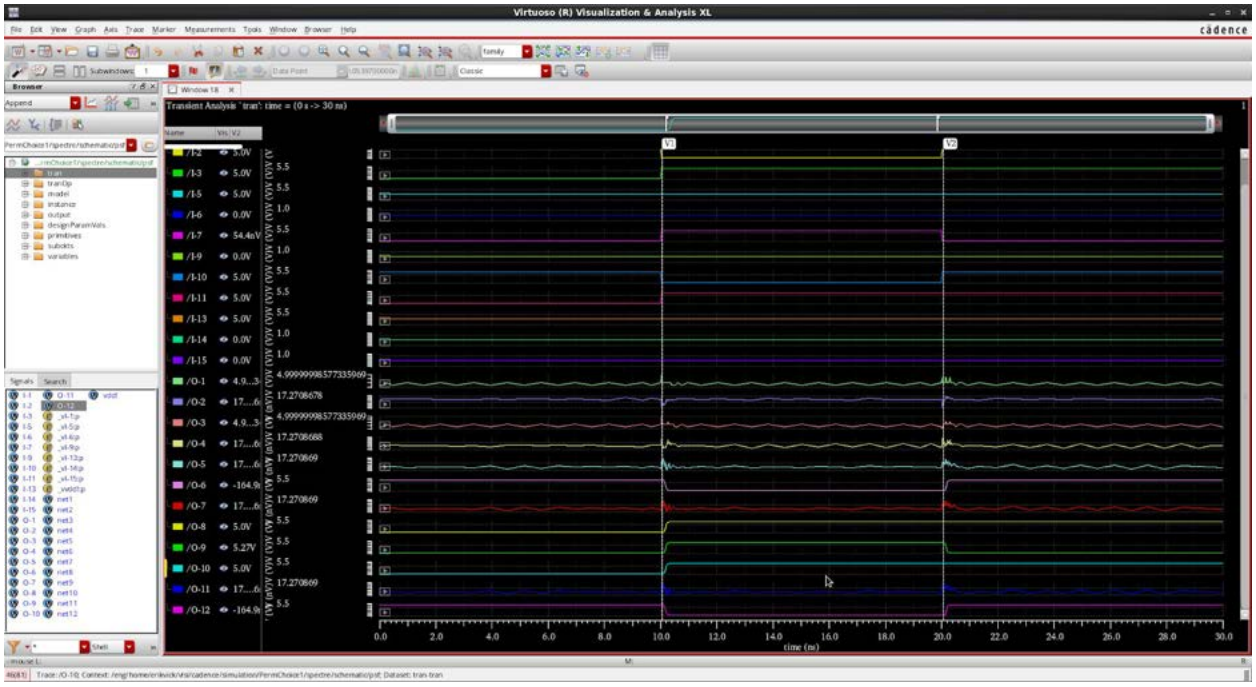


Figure 23: Schematic Simulation of Permutation Choice 1
I-1 is “000” and I-2 is “101”

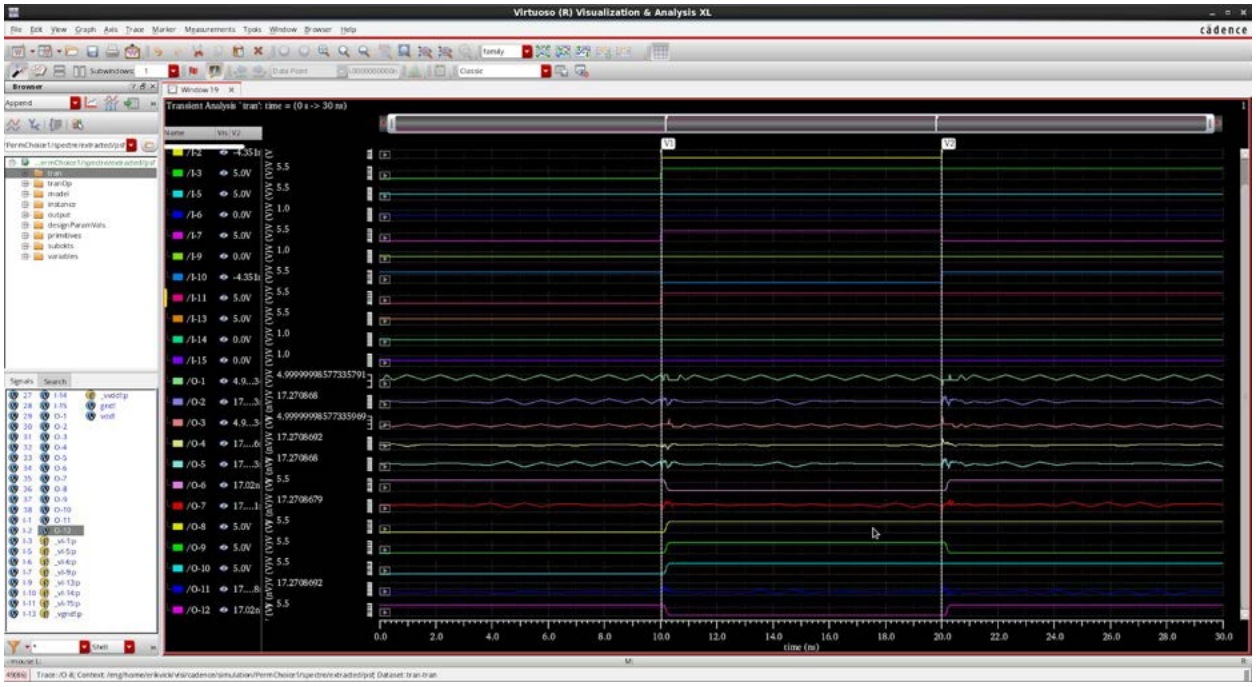


Figure 24: Extracted Simulation of Permutation Choice 1
I-1 is “000” and I-2 is “101”

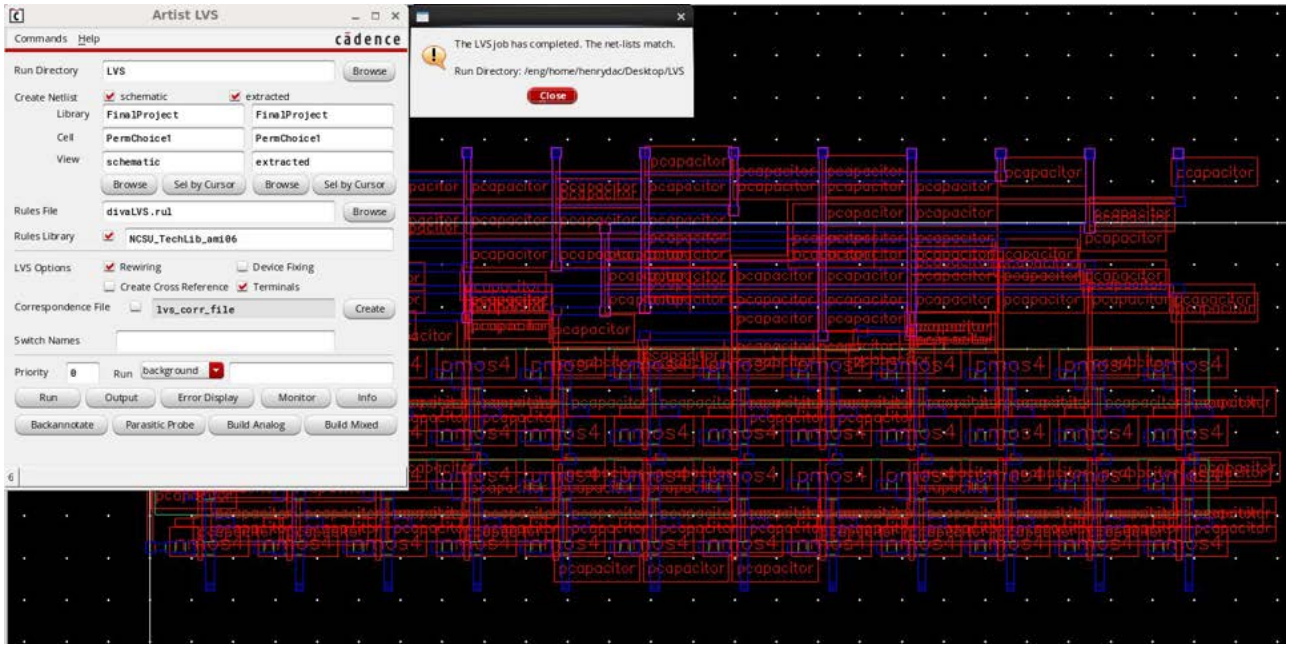


Figure 25: LVS Check for Permutation Choice 1

Permutation Choice 2 - Views:

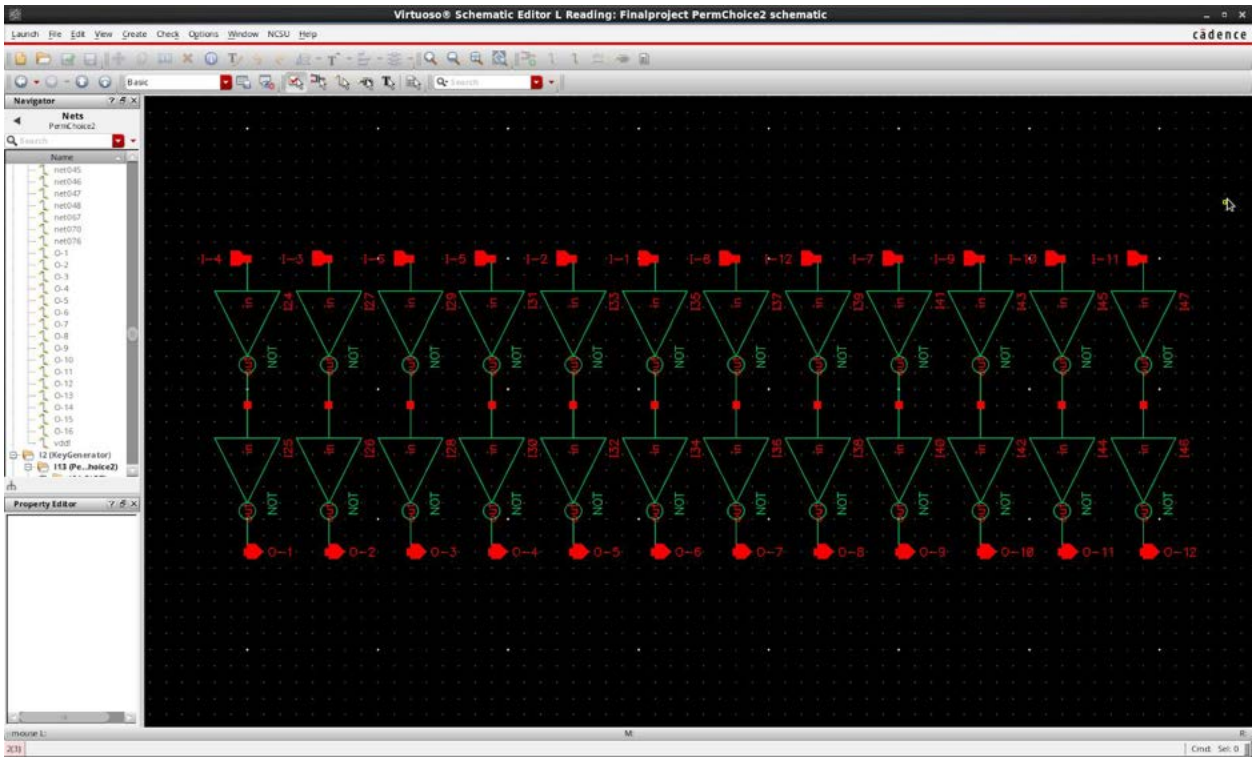


Figure 26: Schematic View of Permutation Choice 2

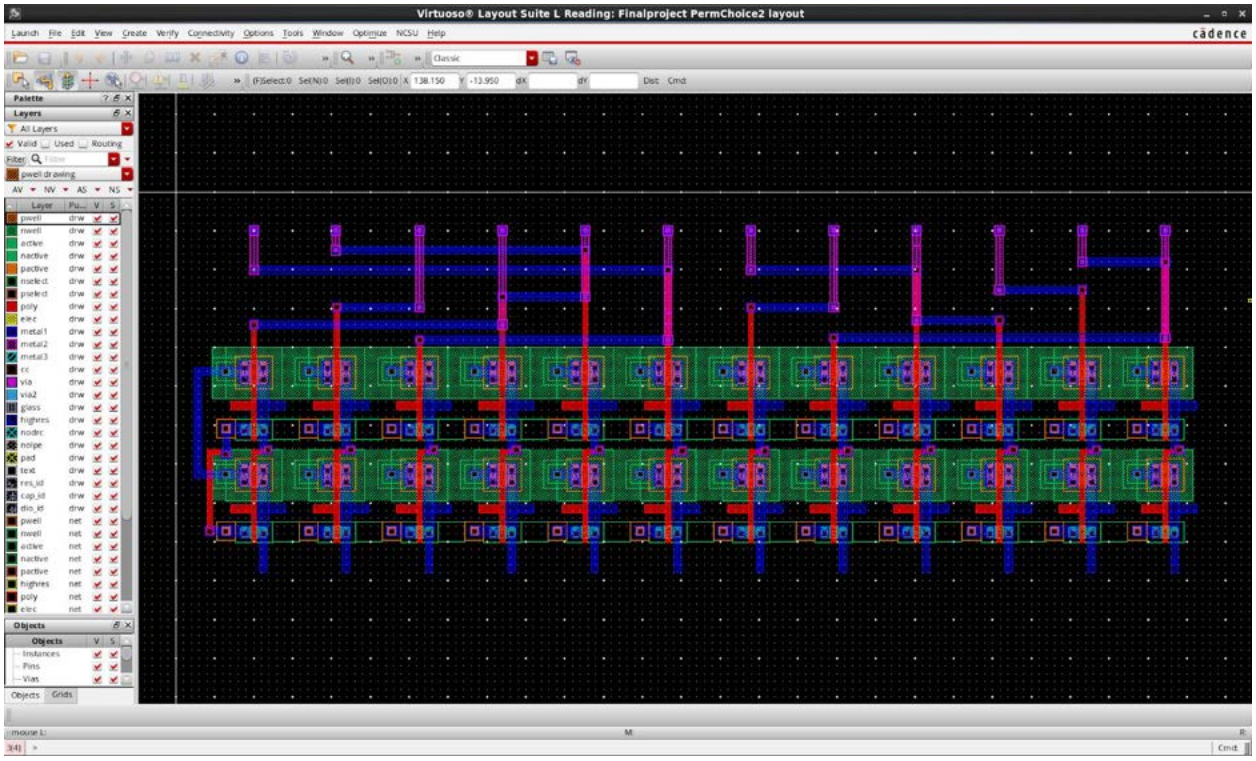


Figure 27: Layout View of Permutation Choice 2

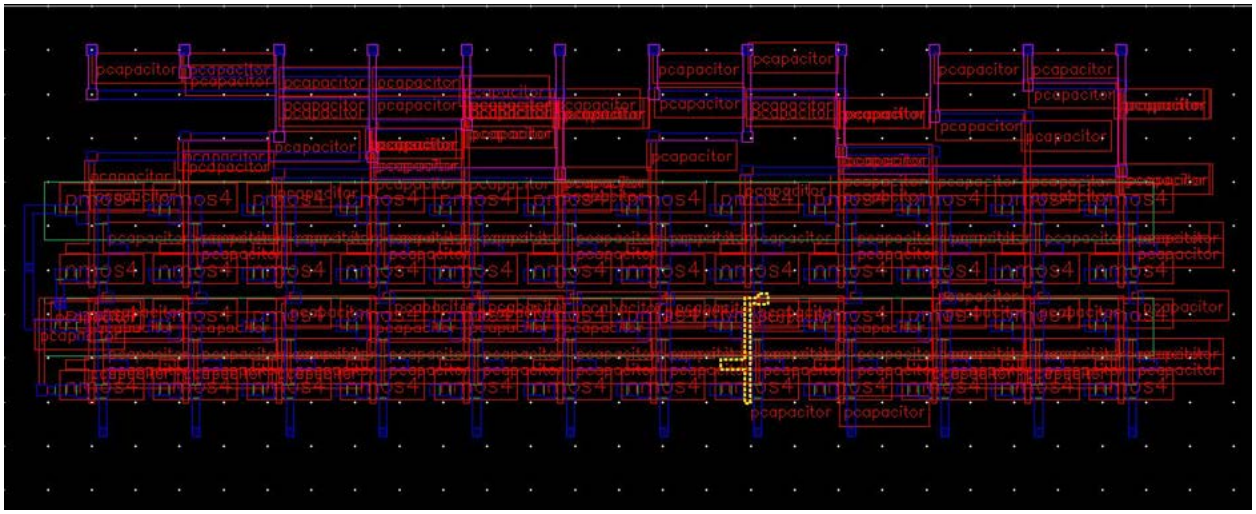


Figure 28: Extracted View of Permutation Choice 2

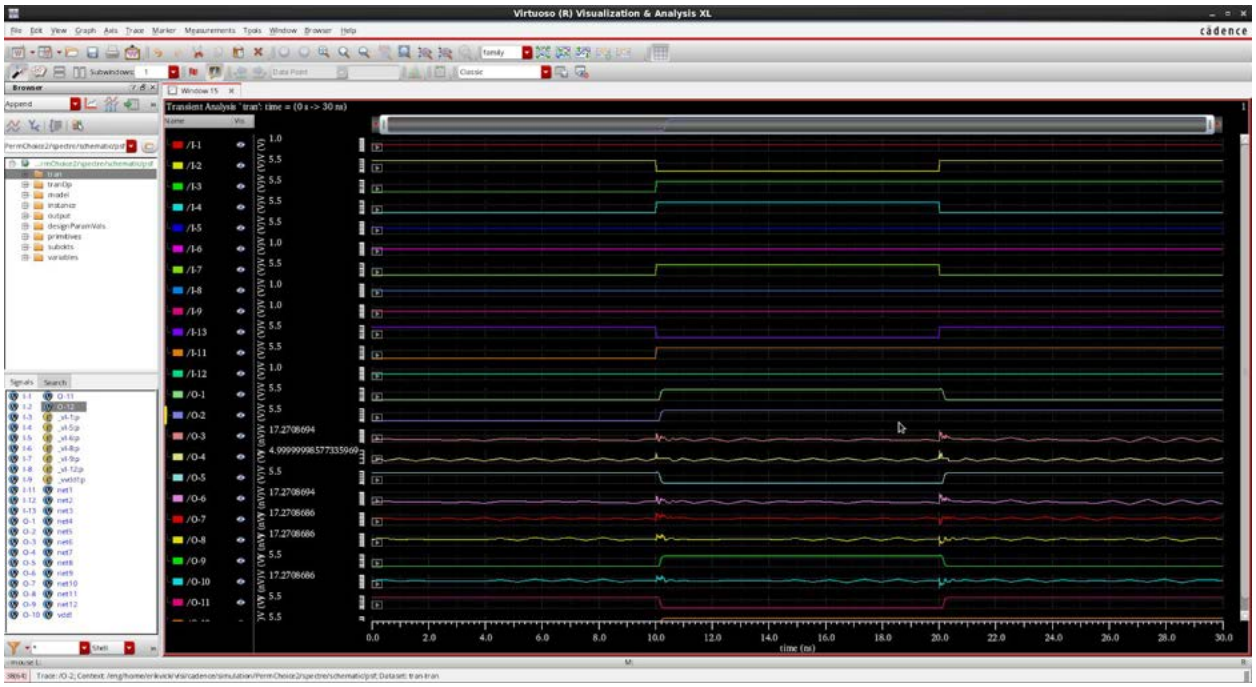


Figure 29: Schematic Simulation of Permutation Choice 2
O-12 has a value of "011"

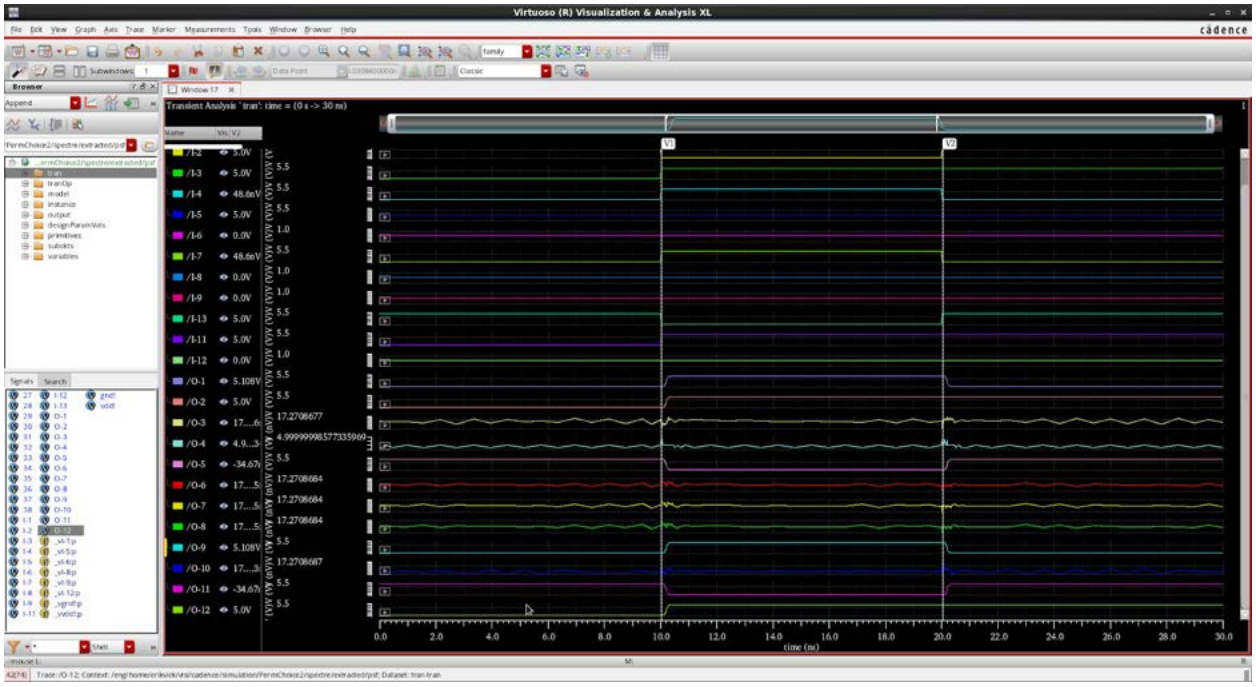


Figure 30: Extracted Simulation of Permutation Choice 2
Note: I-1 has a value of "000"

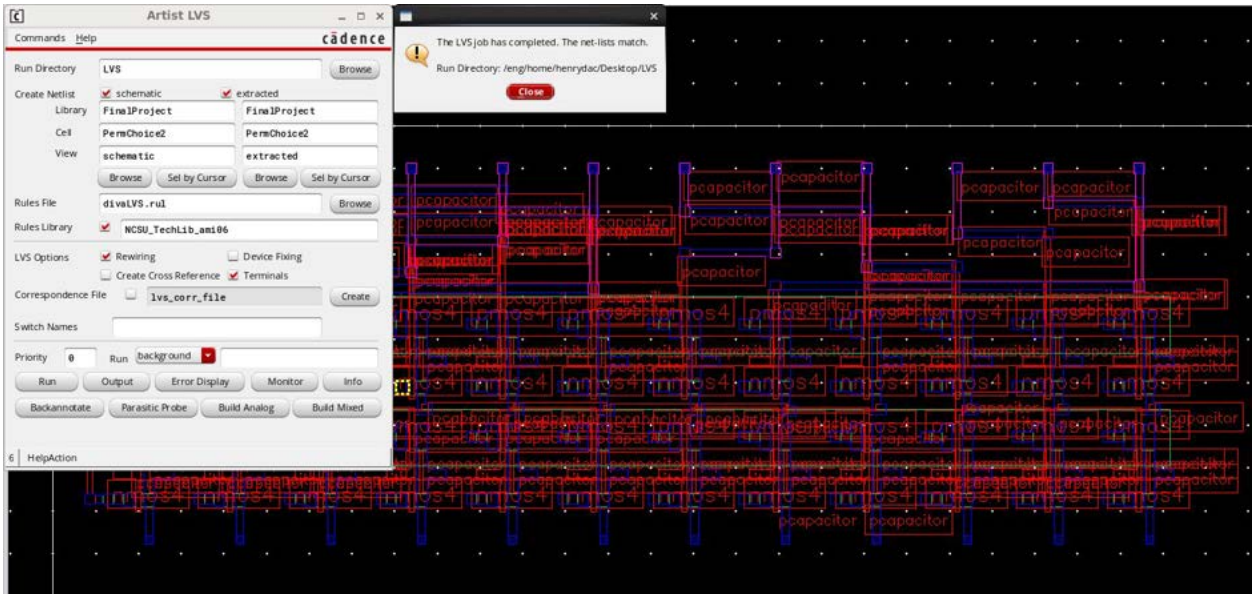


Figure 31: LVS Check for Permutation Choice 2

Expansion Function - Views:

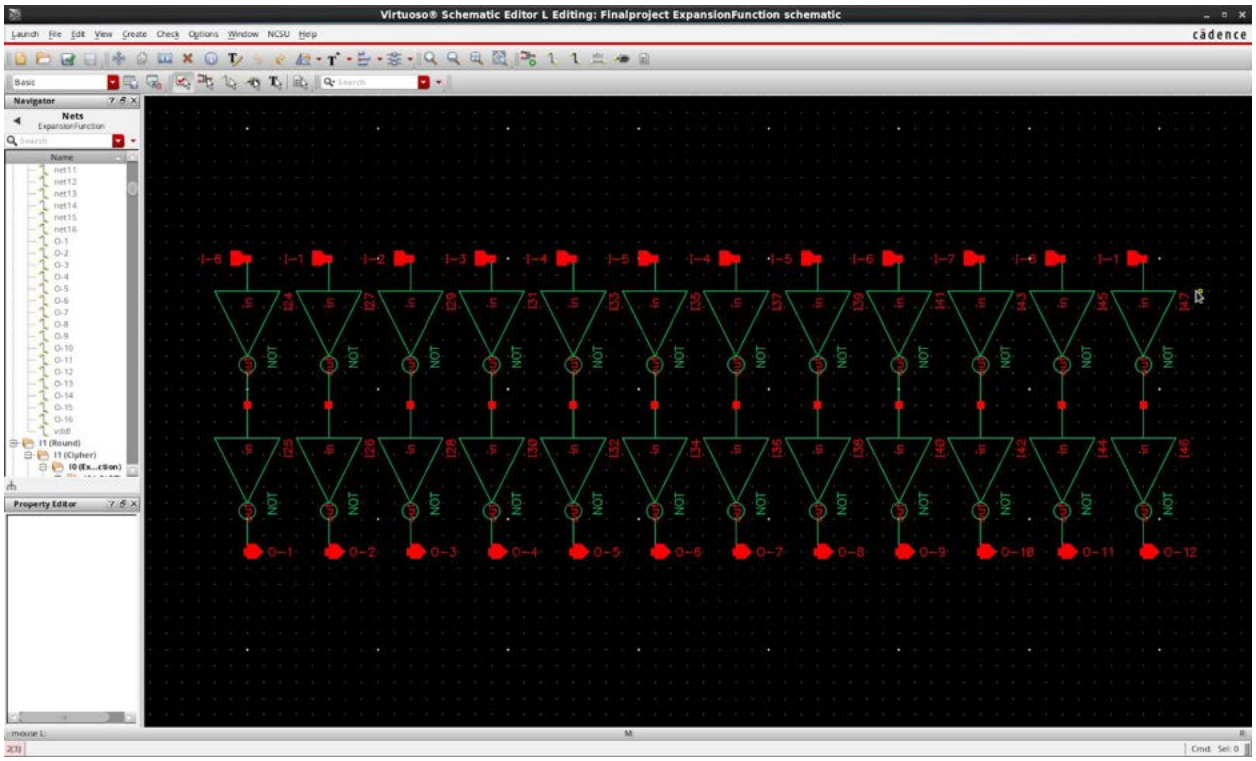


Figure 32: Schematic View of the Expansion Function

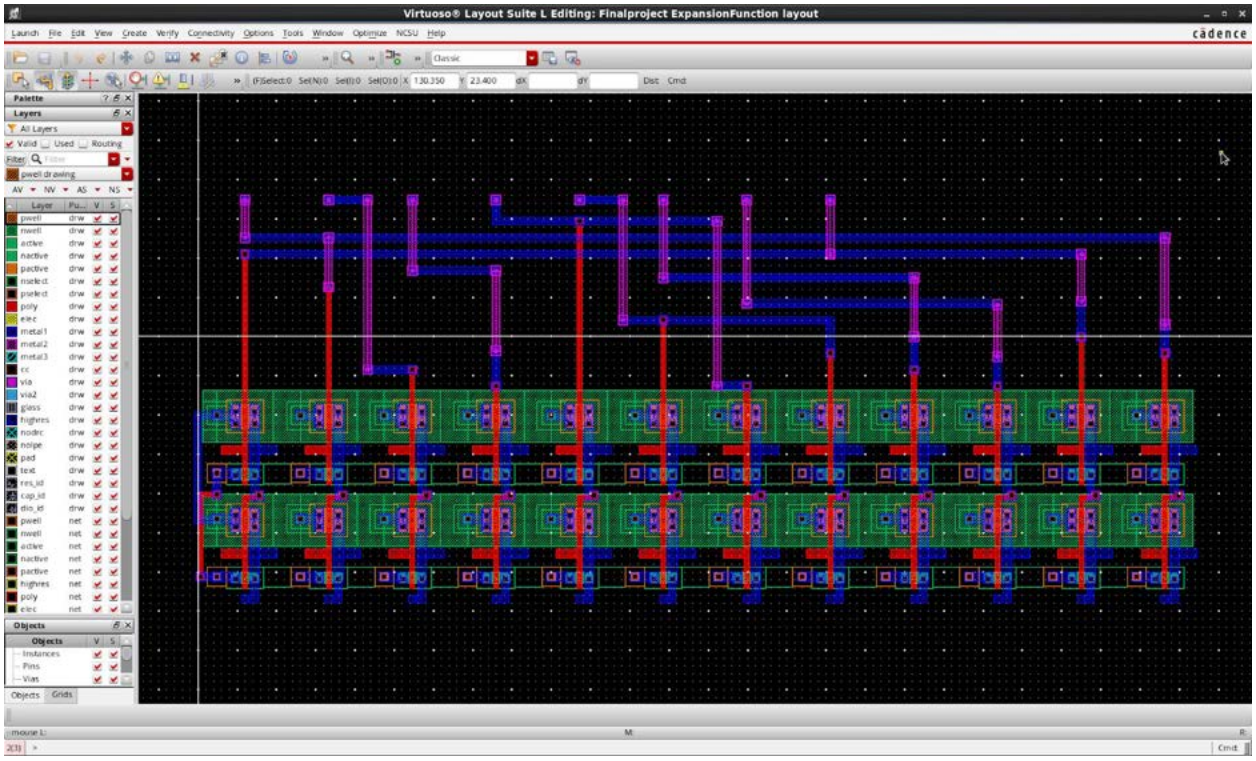


Figure 33: Layout View of the Expansion Function

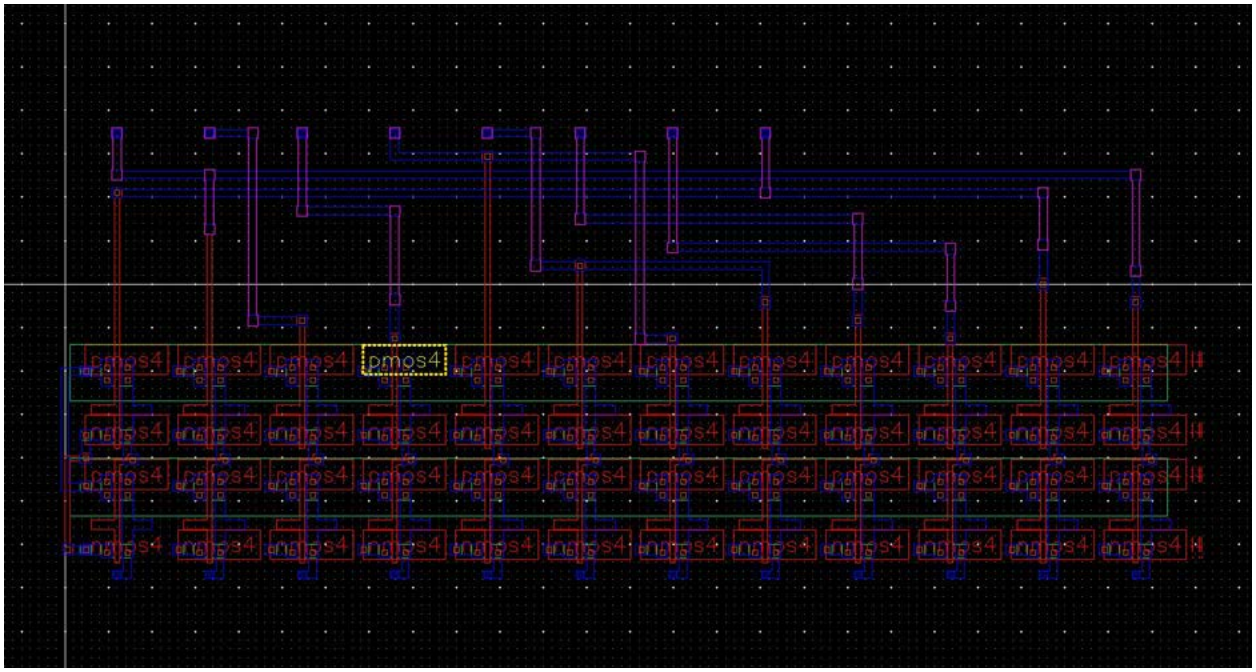


Figure 34: Extracted View of the Expansion Function

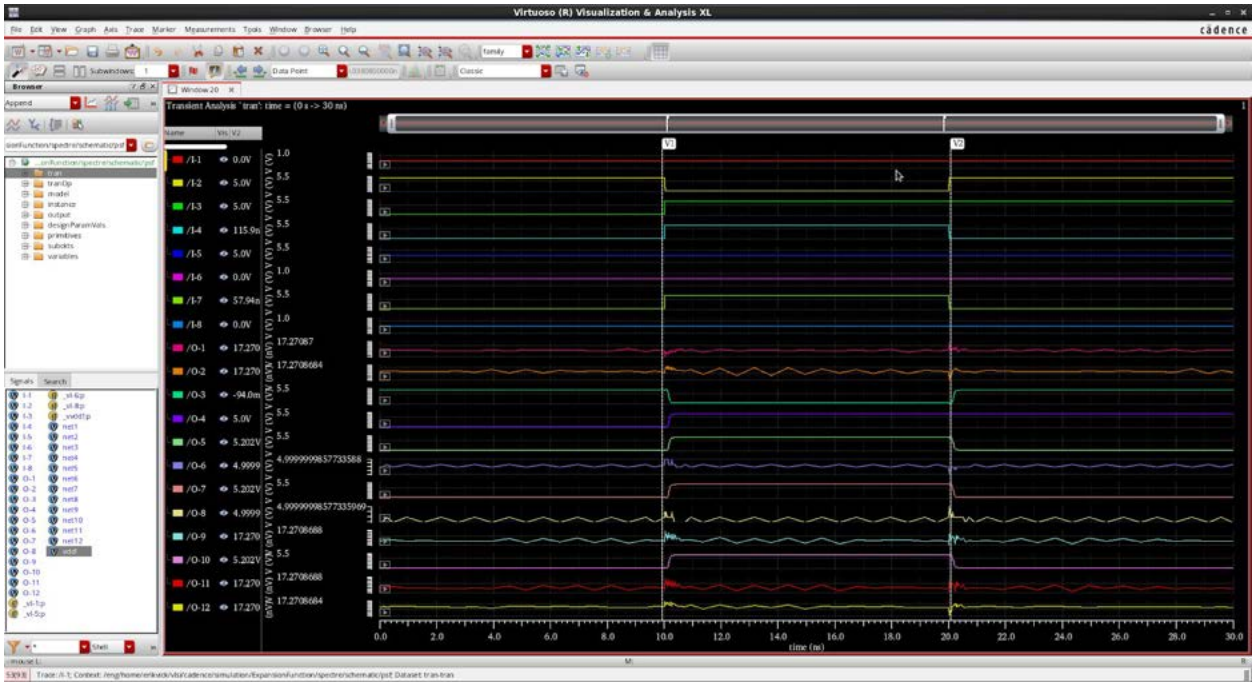


Figure 35: Schematic Simulation of the Expansion Function

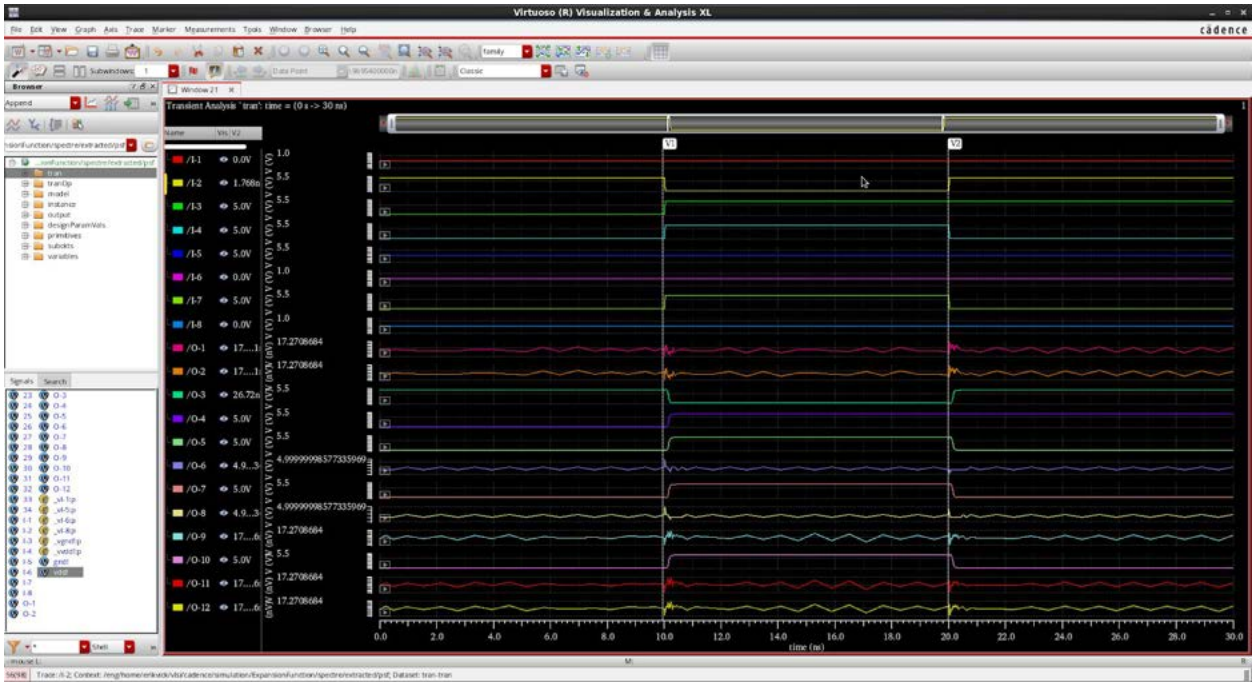


Figure 36: Extracted Simulation of the Expansion Function

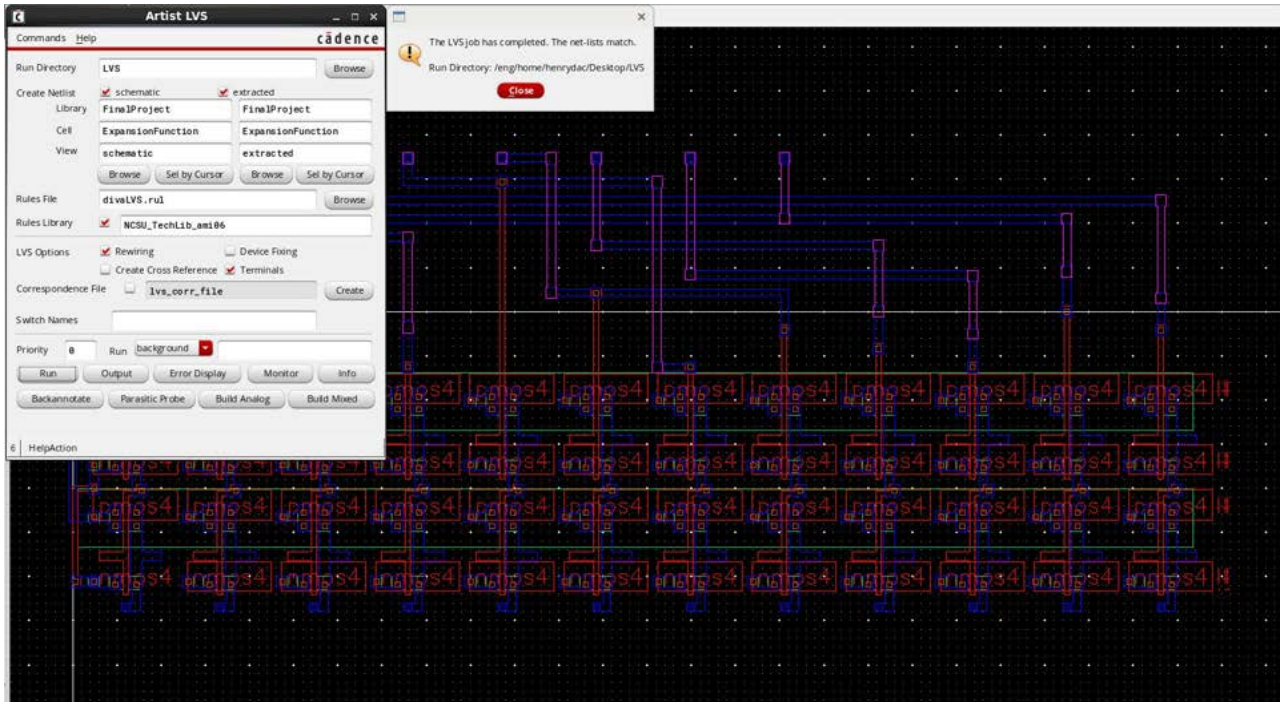


Figure 37: LVS Check for the Expansion Function

Circular Shift Left - 1bit - Views:

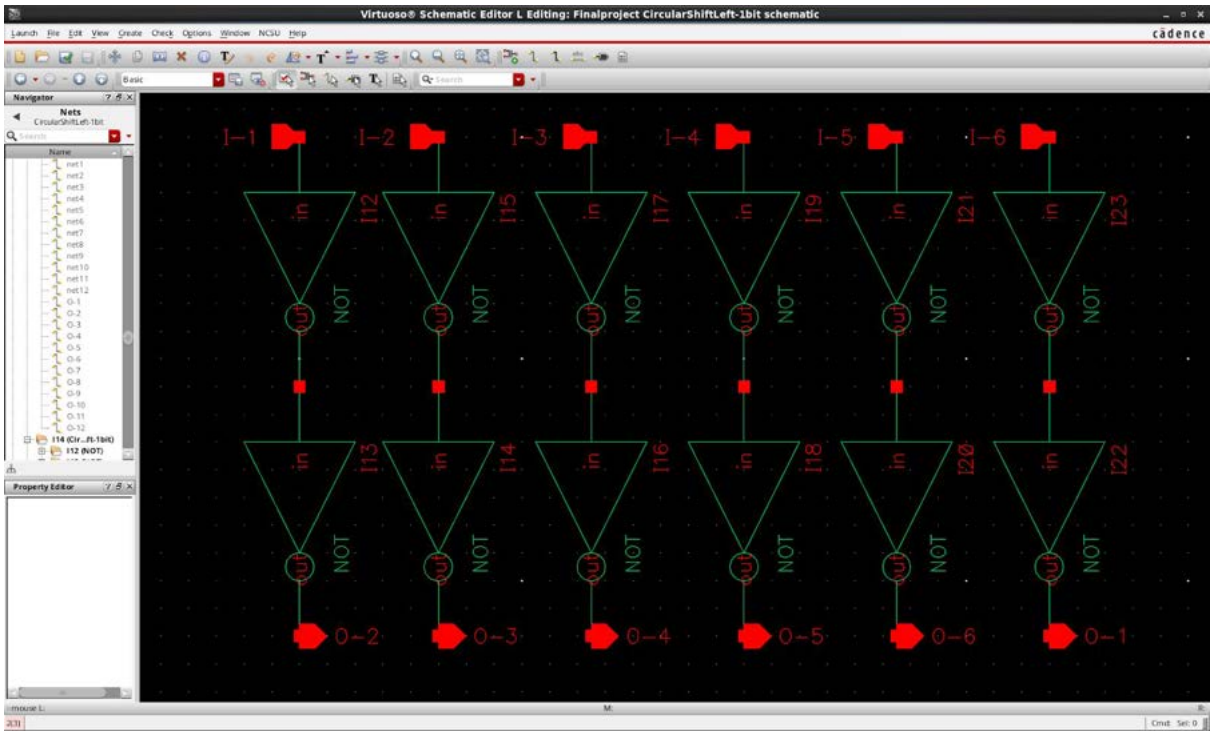


Figure 38: Schematic View of the Circular Shift Left - 1bit

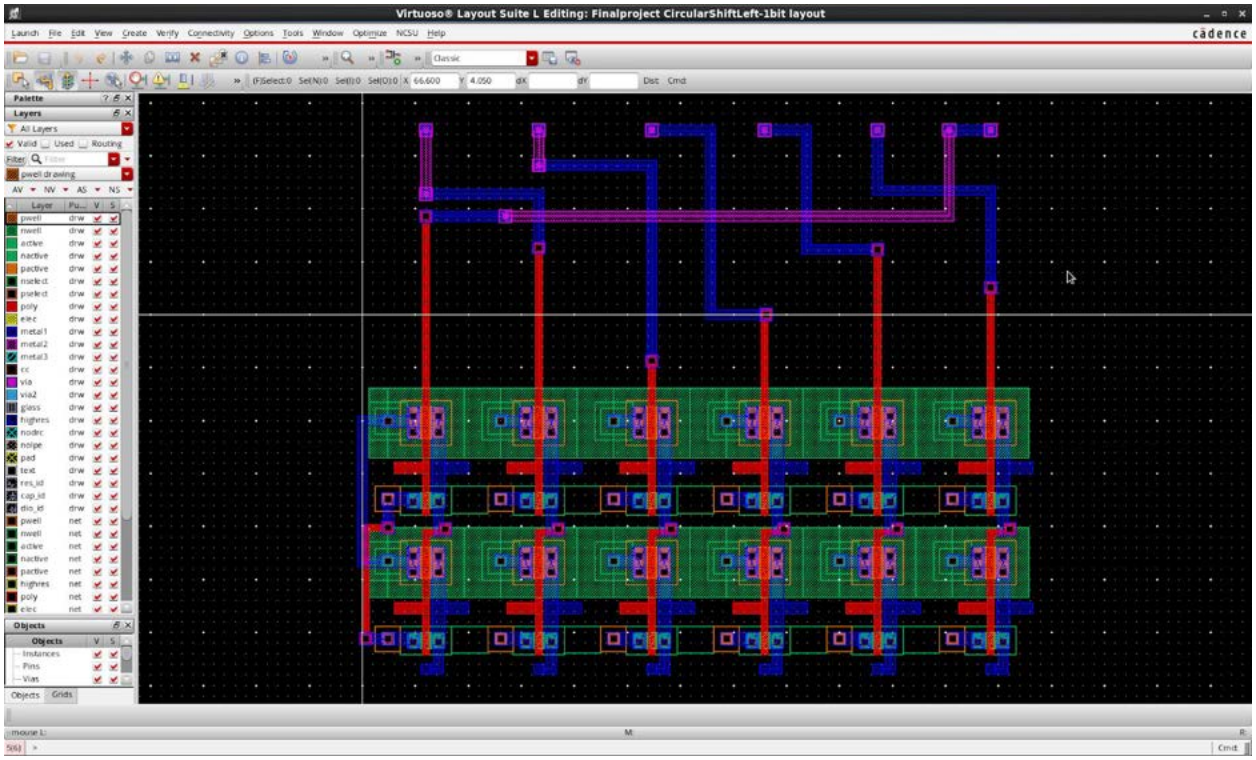


Figure 39: Layout View of the Circular Shift Left - 1bit

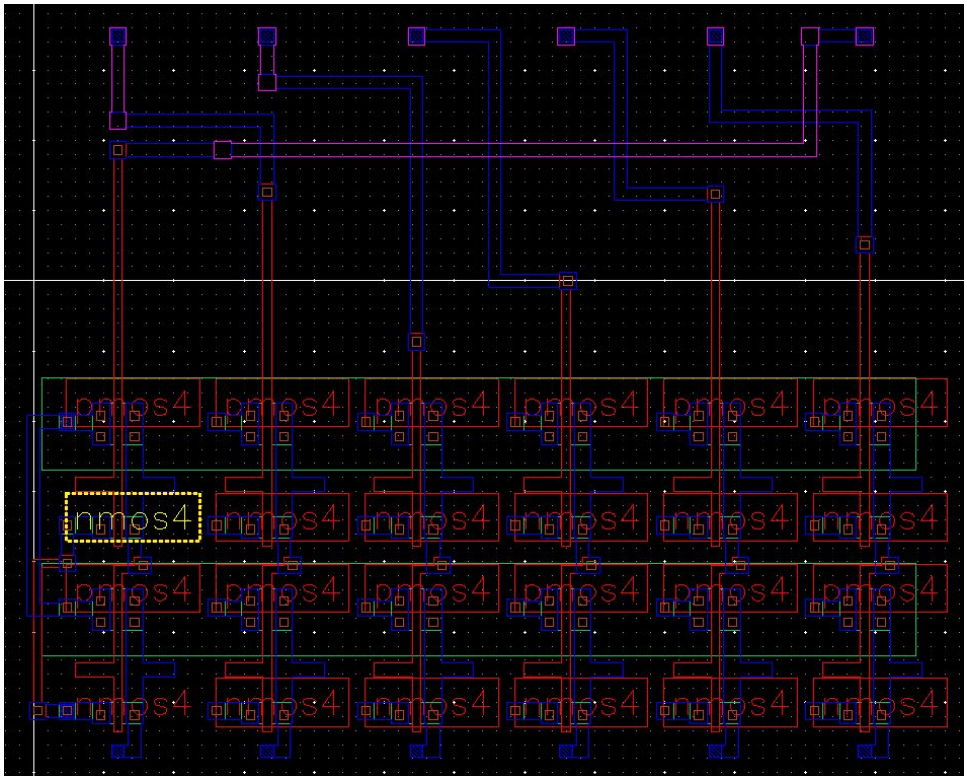


Figure 40: Extracted View of the Circular Shift Left - 1bit

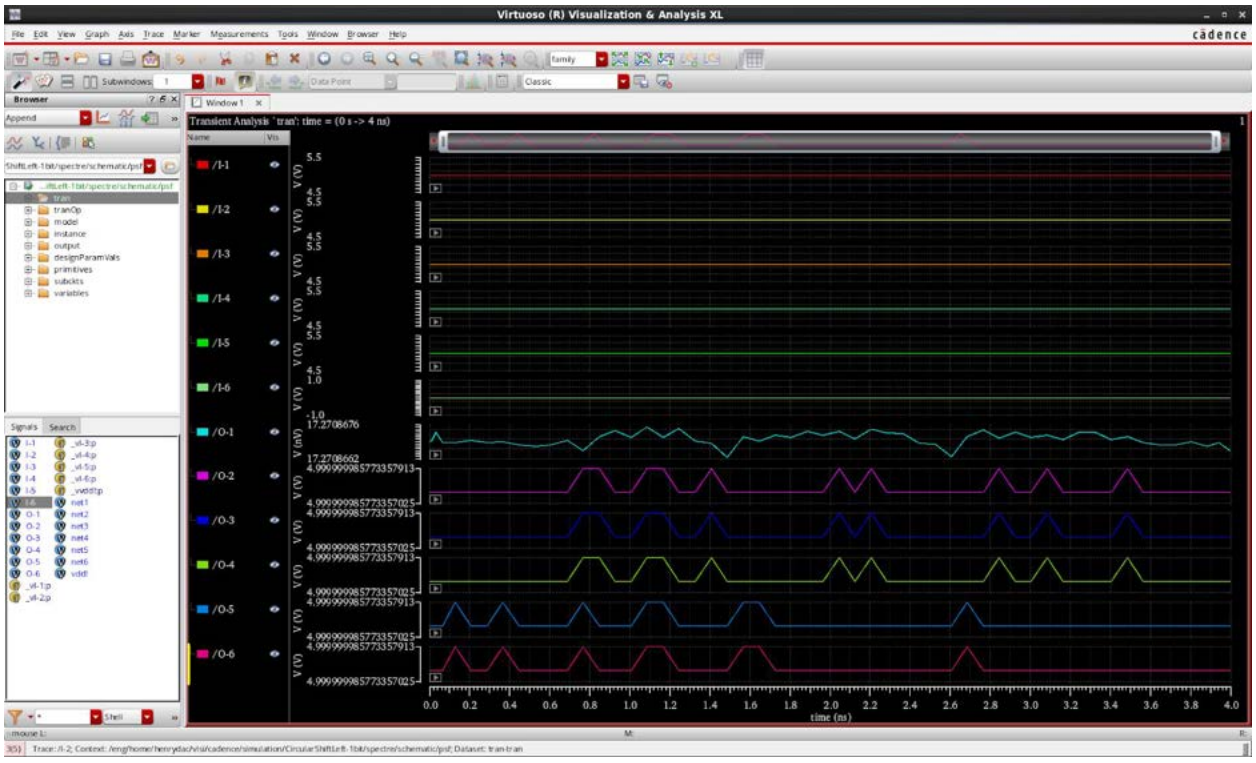


Figure 41: Schematic Simulation of the Circular Shift Left - 1bit

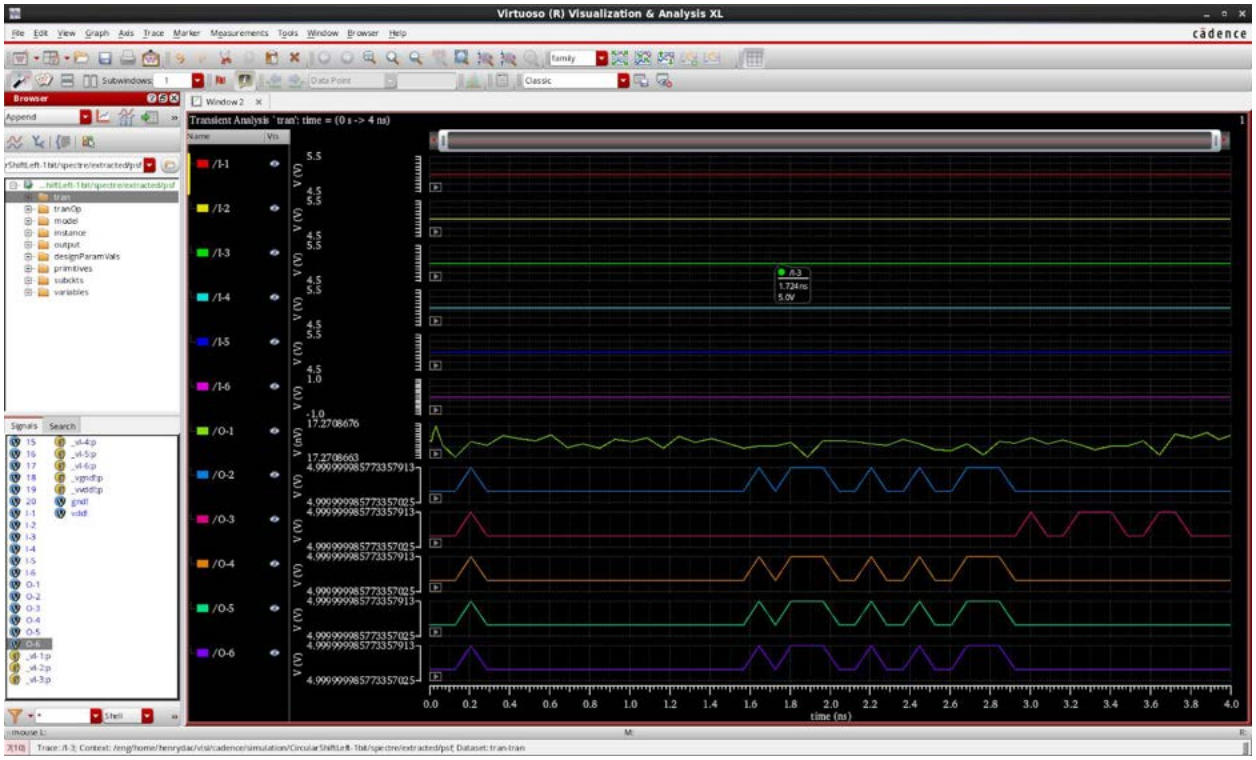


Figure 42: Extracted Simulation of the Circular Shift Left - 1bit

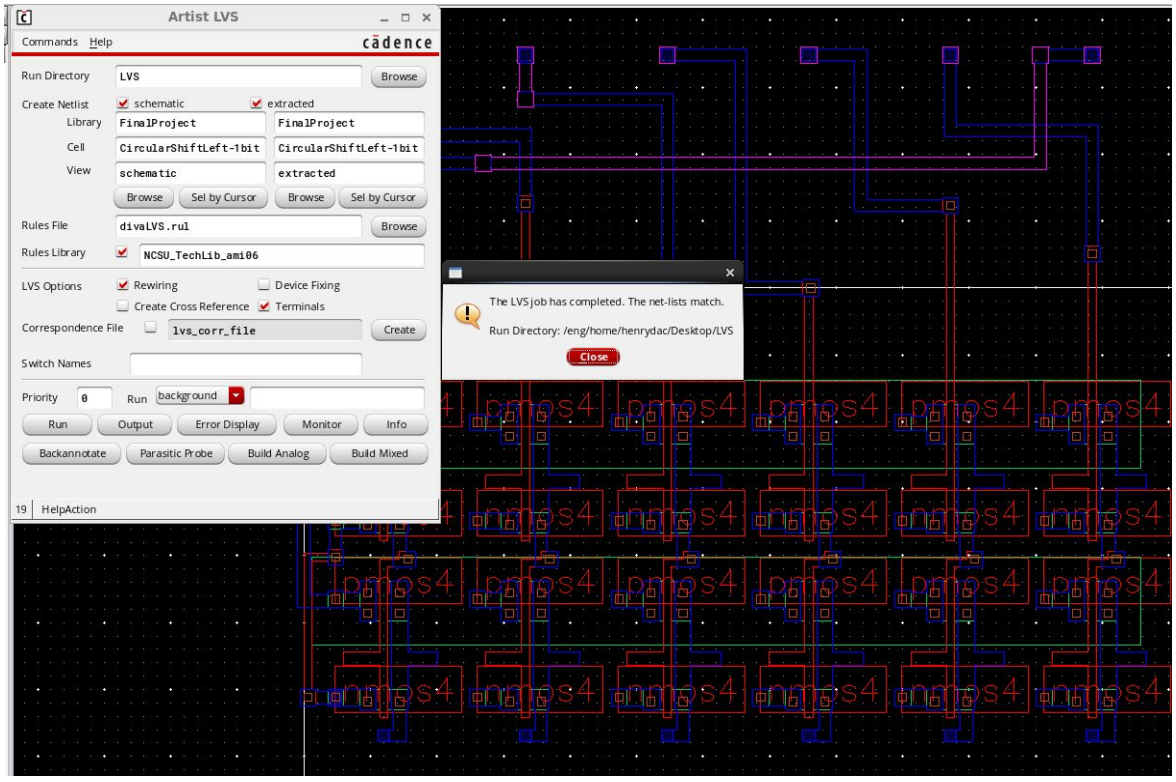


Figure 43: LVS Check for the Circular Shift Left - 1bit

Circular Shift Left - 2bit - Views:

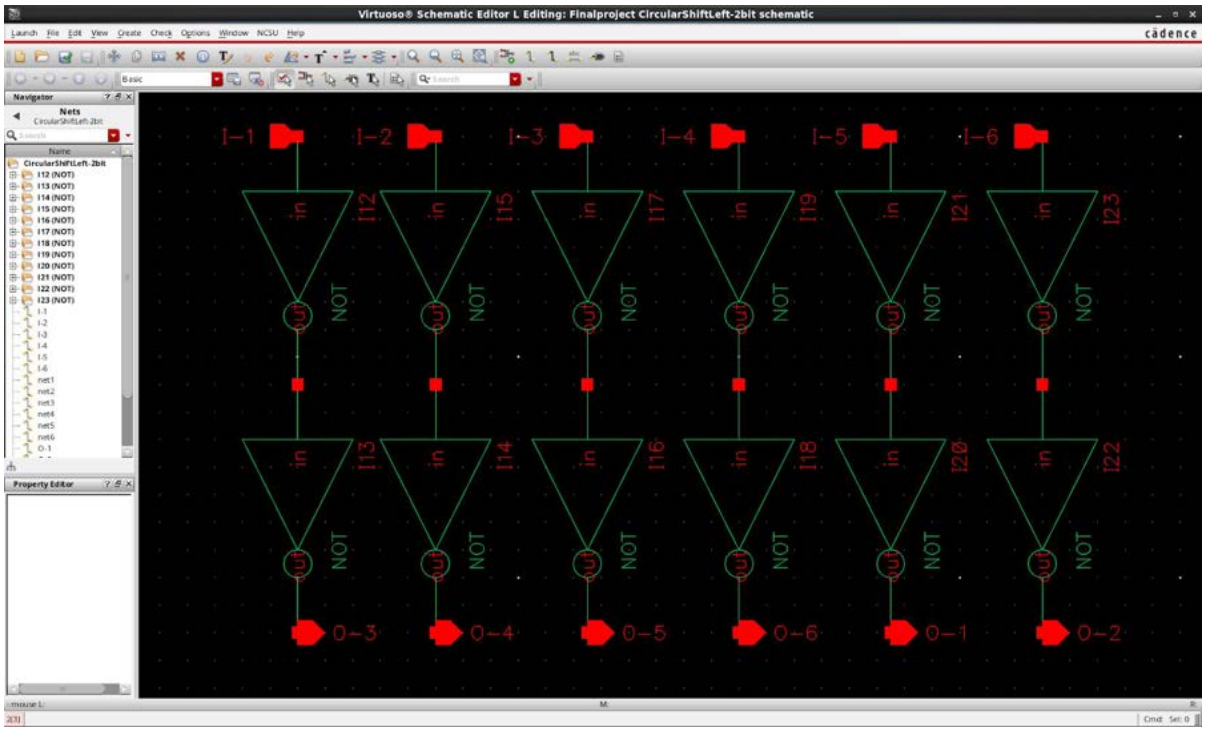


Figure 44: Schematic View of the Circular Shift Left - 2bit

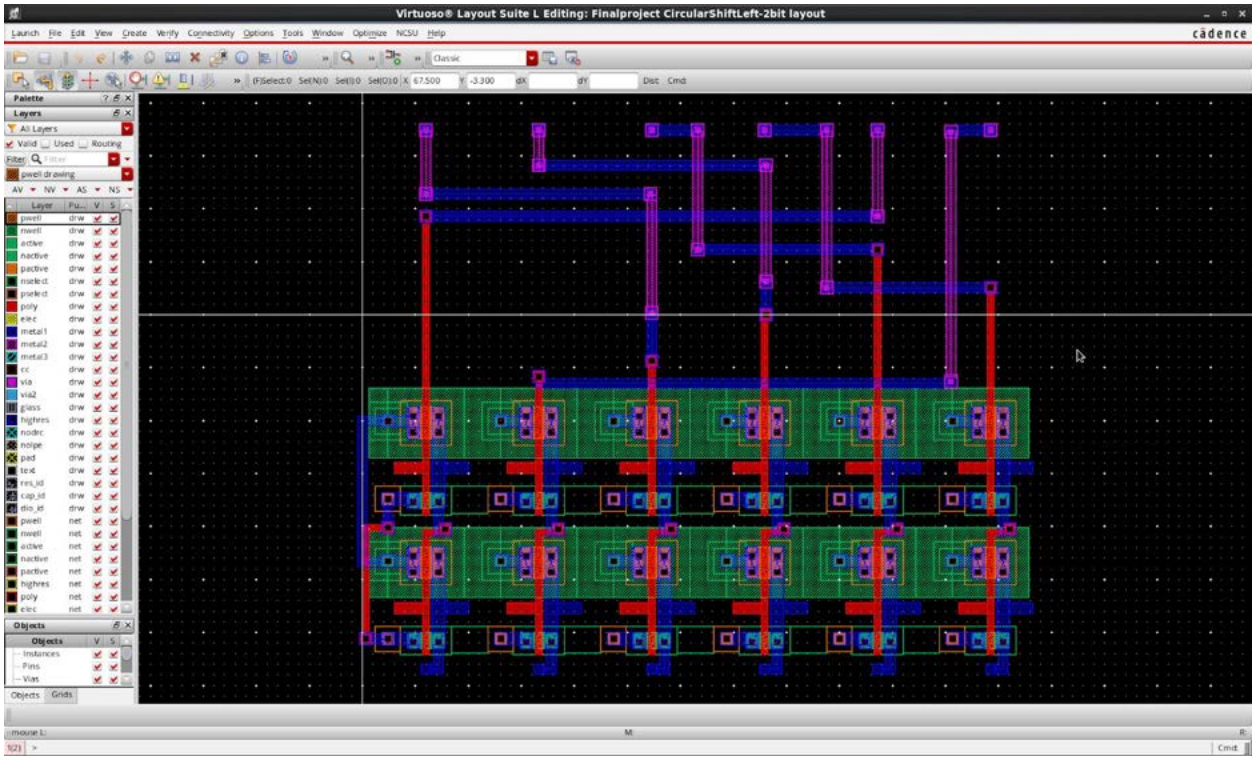


Figure 45: Layout View of the Circular Shift Left - 2bit

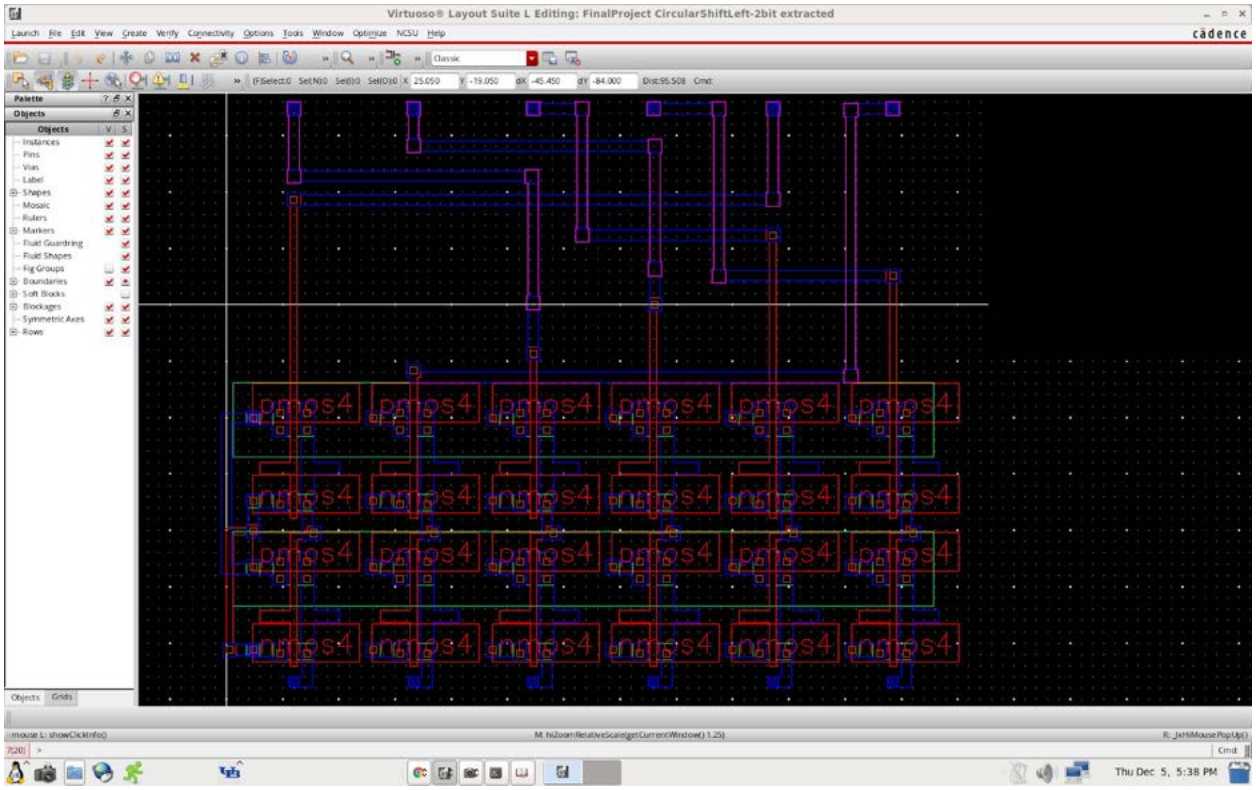


Figure 46: Extracted View of the Circular Shift Left - 2bit

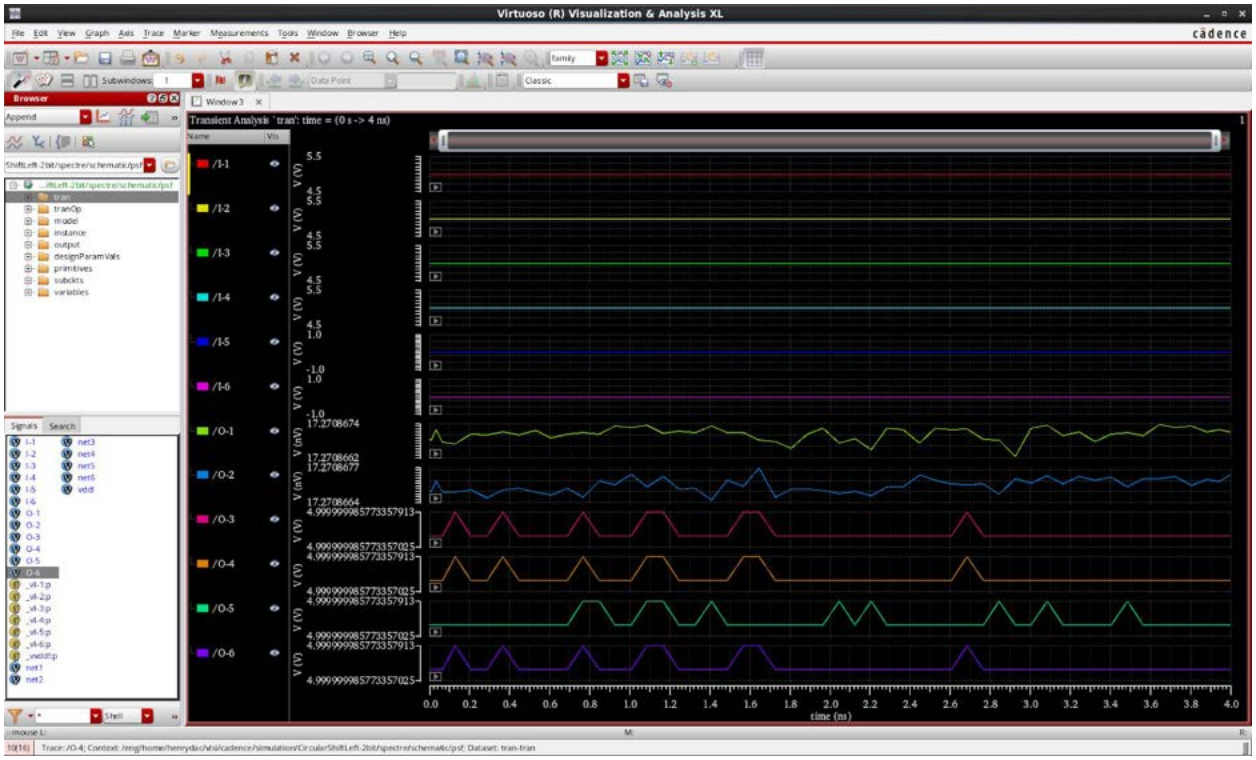


Figure 47: Schematic Simulation of the Circular Shift Left - 2bit

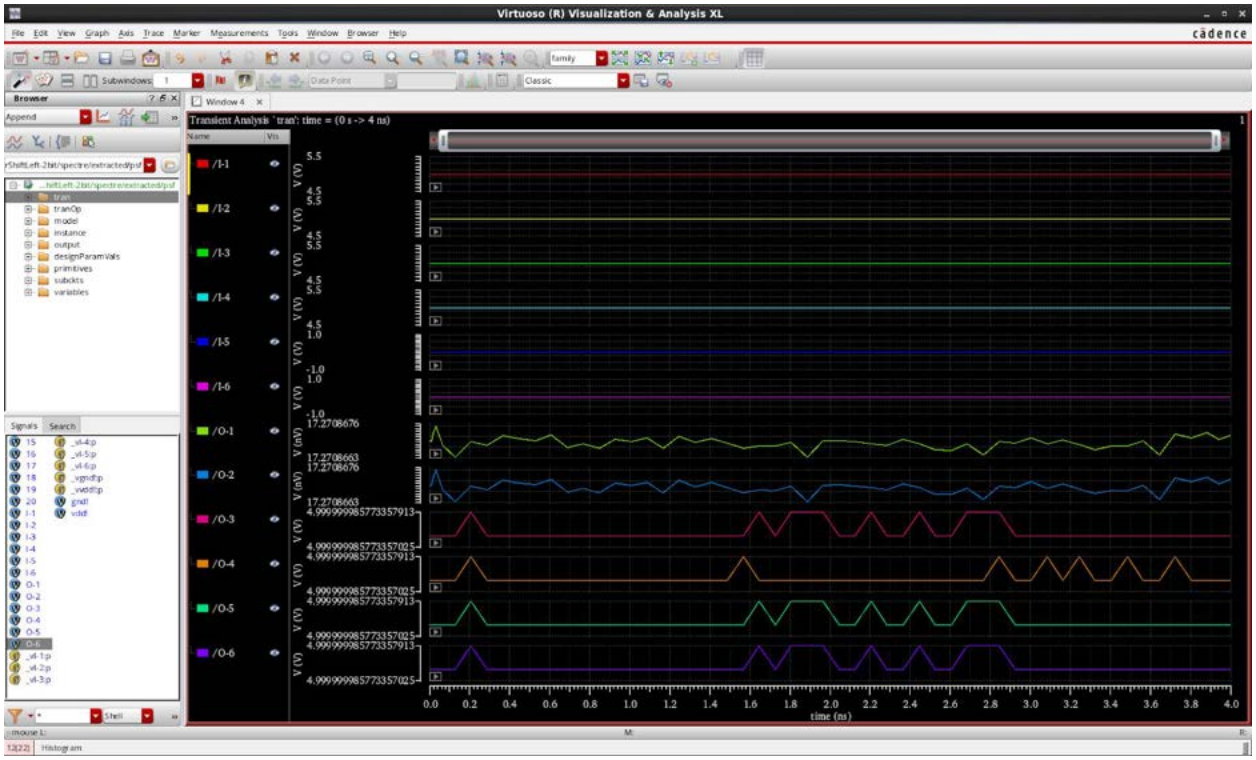


Figure 48: Extracted Simulation of the Circular Shift Left - 2bit

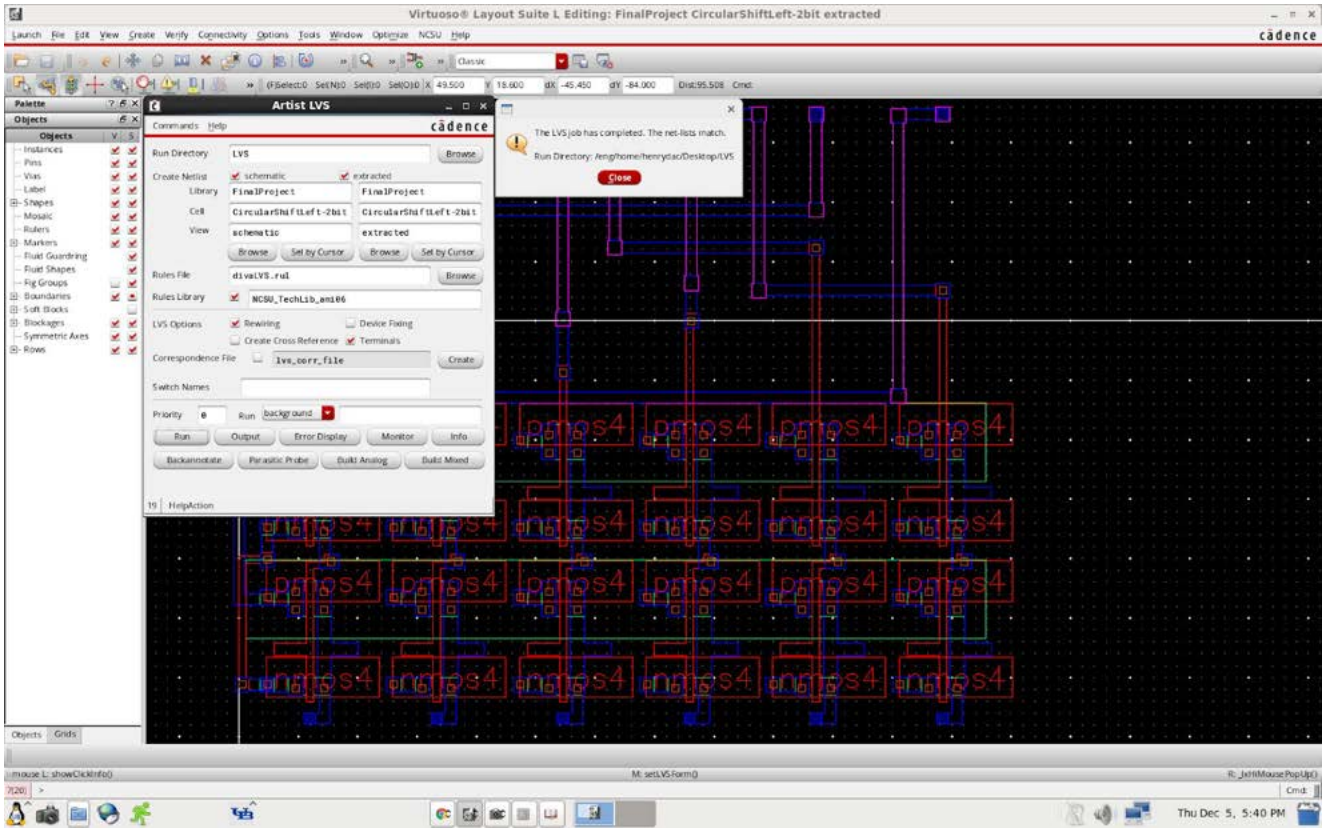


Figure 49: LVS Check for the Circular Shift Left - 2bit

S-box 1 - Views:

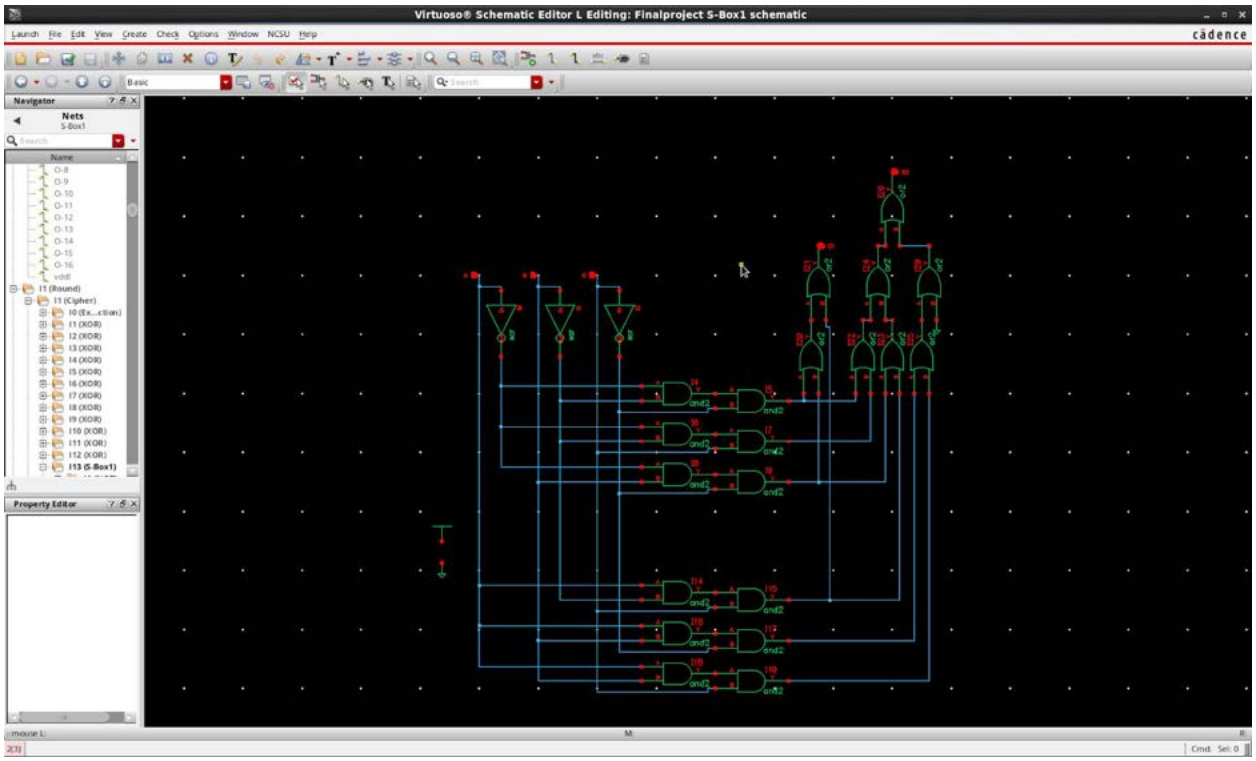


Figure 50: Schematic View of S-box 1

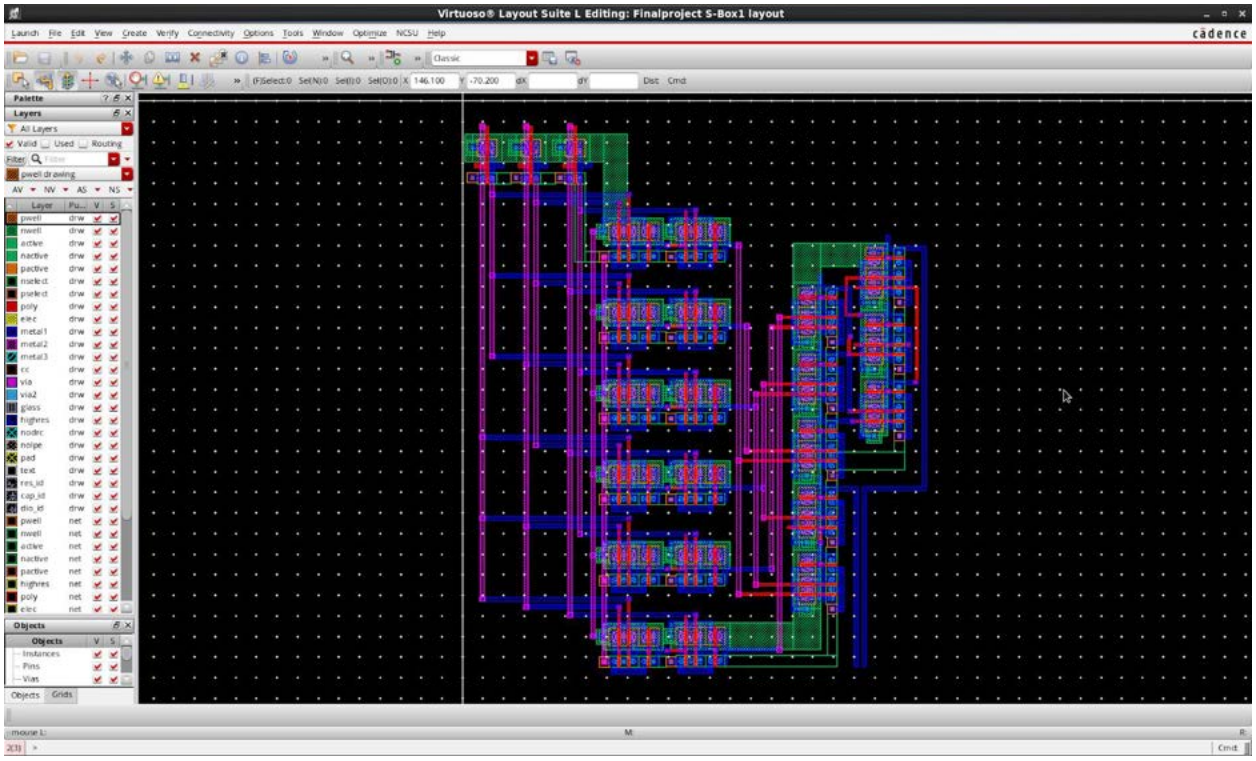


Figure 51: Layout View of S-box 1

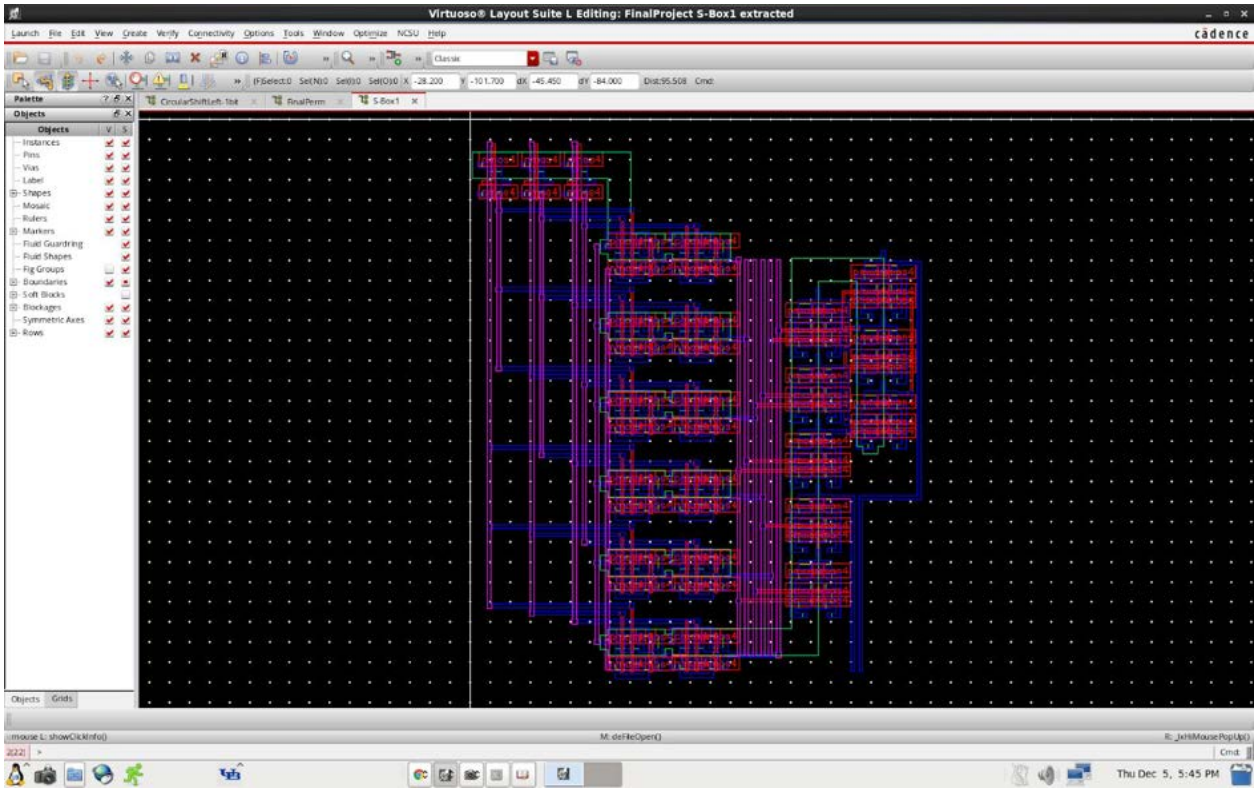


Figure 52: Extracted View of S-box 1

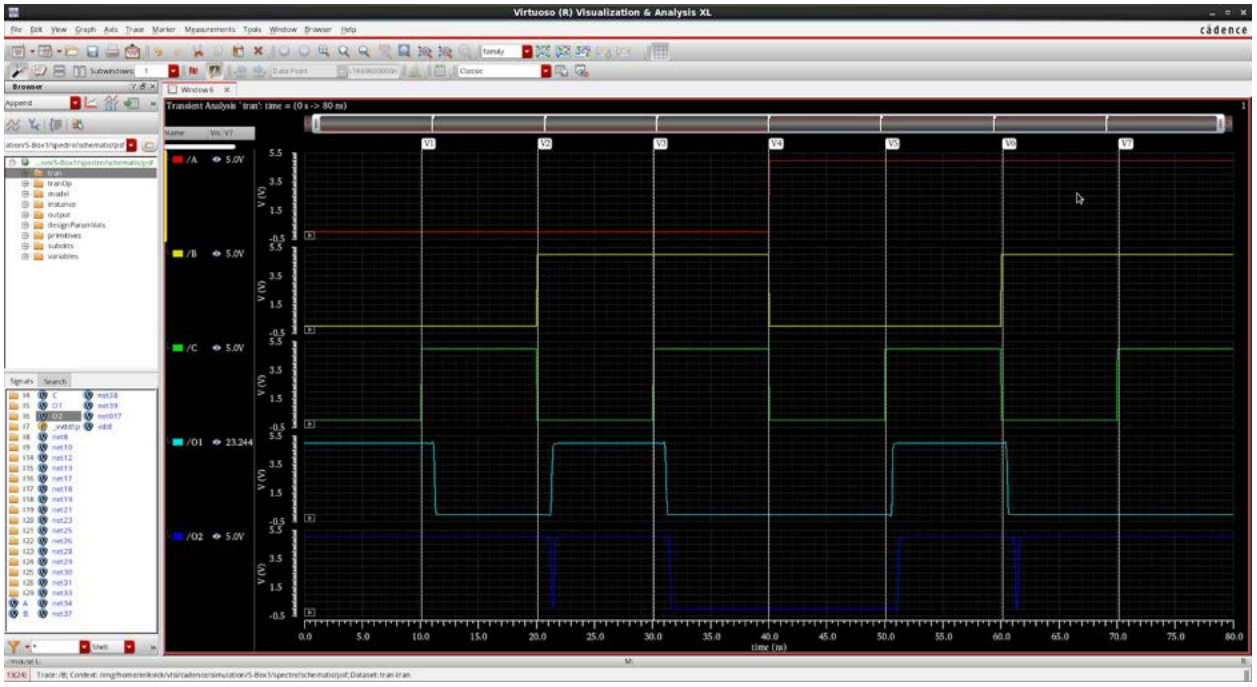


Figure 53: Schematic Simulation of S-box 1

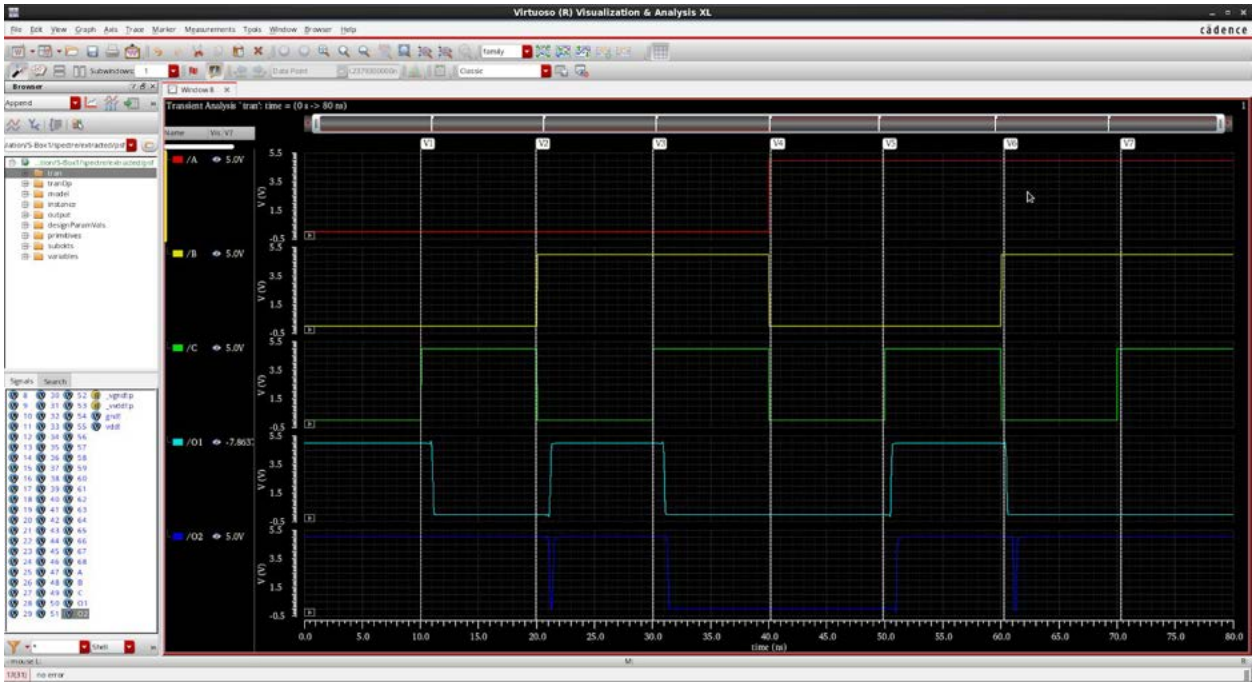


Figure 54: Extracted Simulation of S-box 1

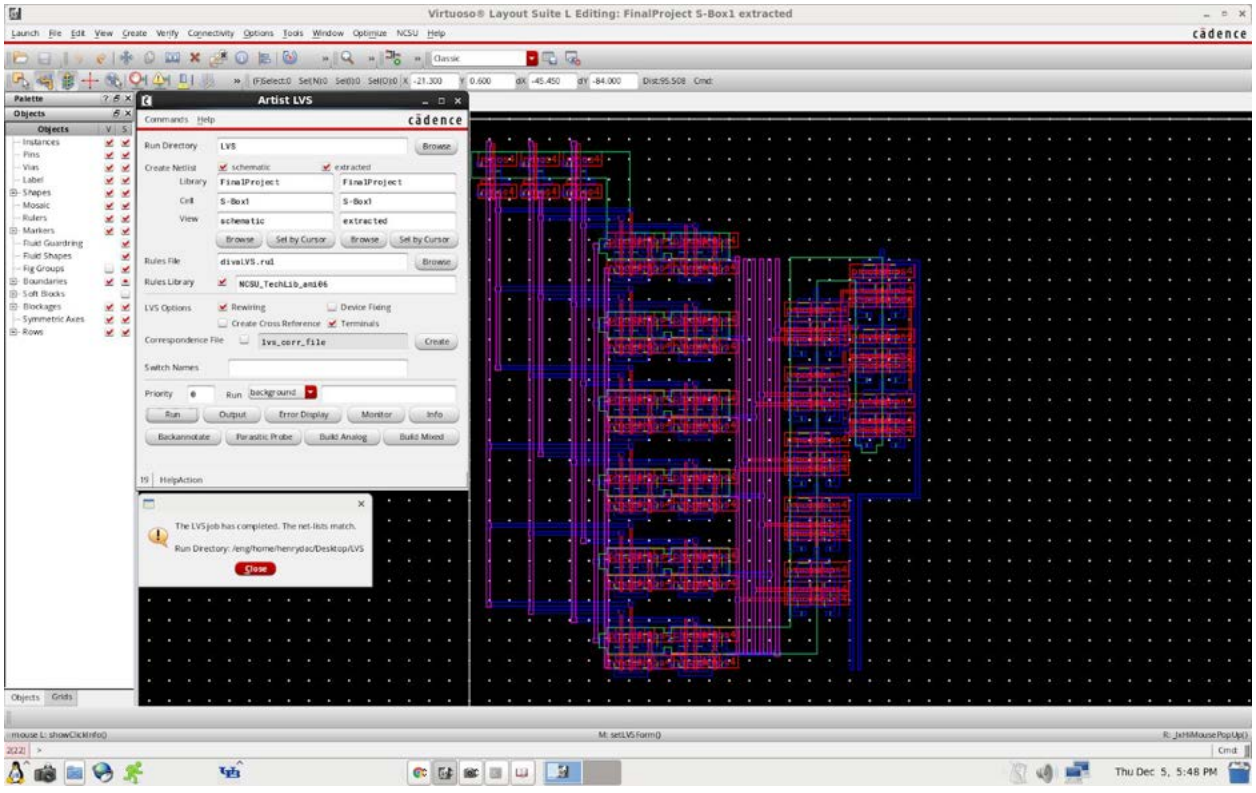


Figure 55: LVS Check for S-box 1

S-box 2 - Views:

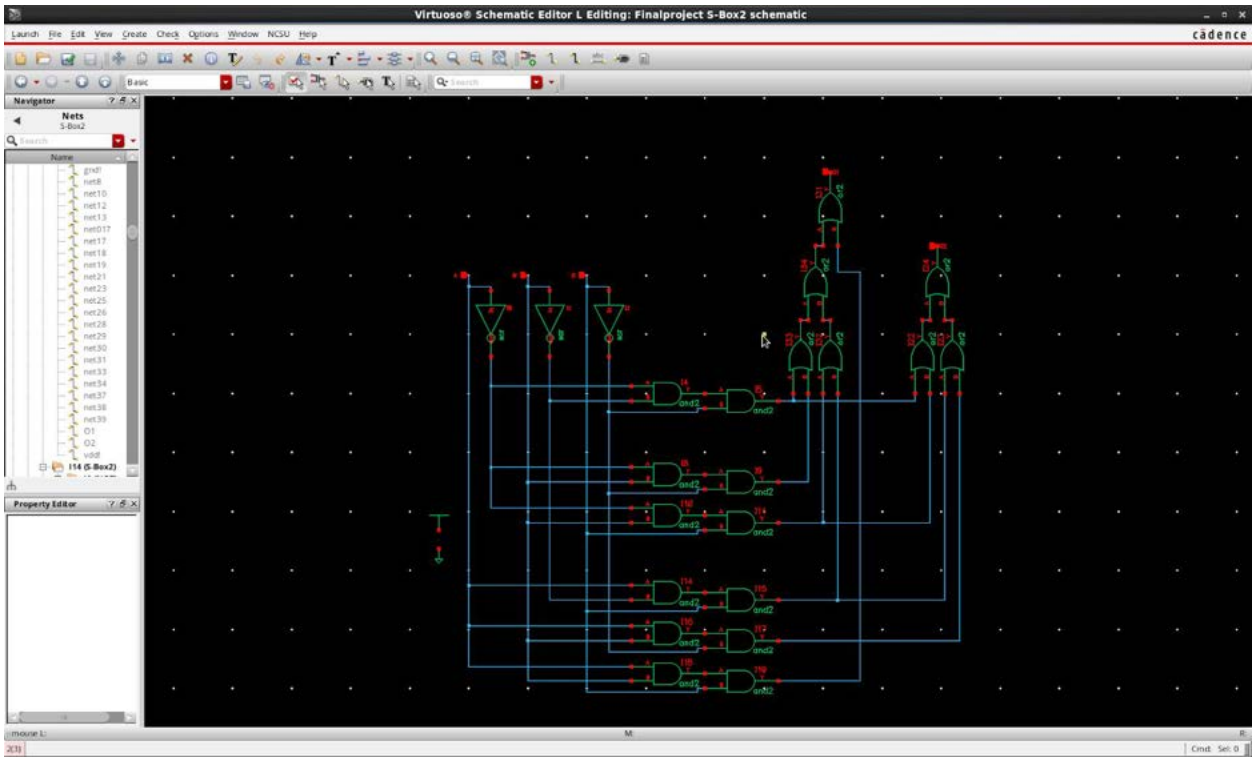


Figure 56: Schematic View of S-box 2

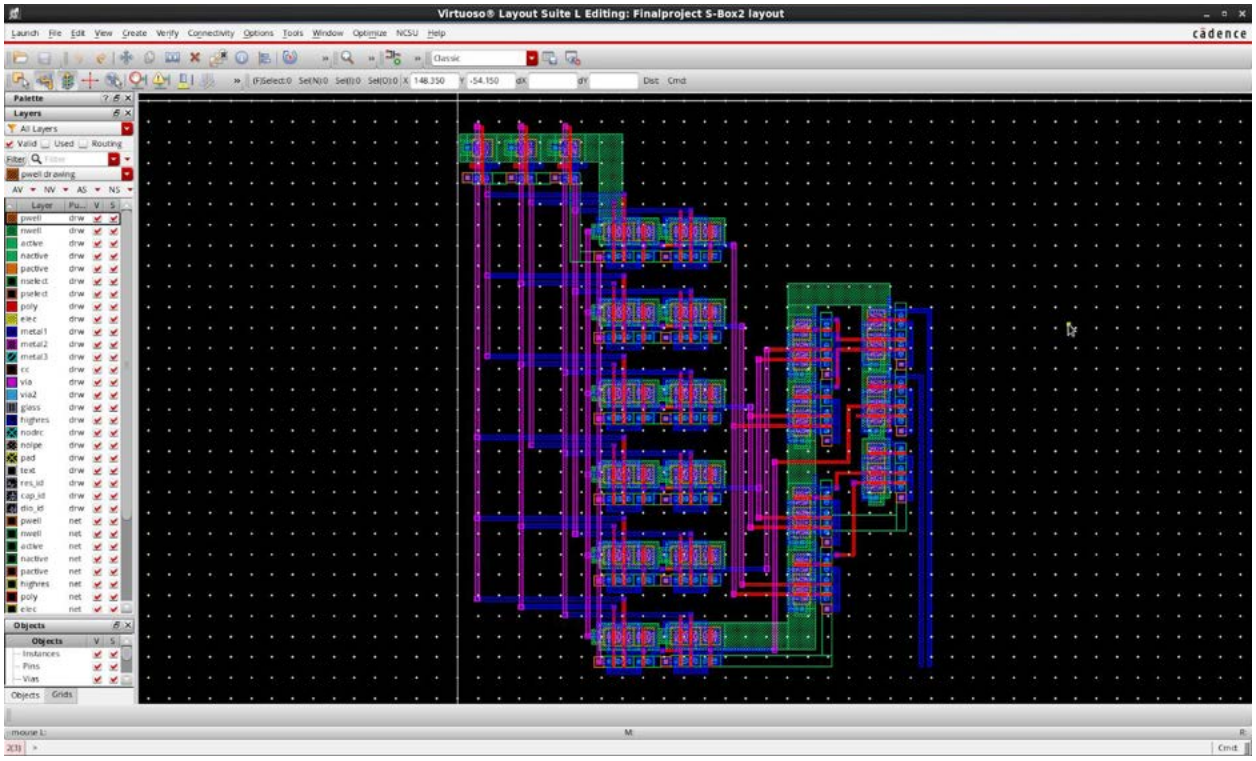


Figure 57: Layout View of S-box 2

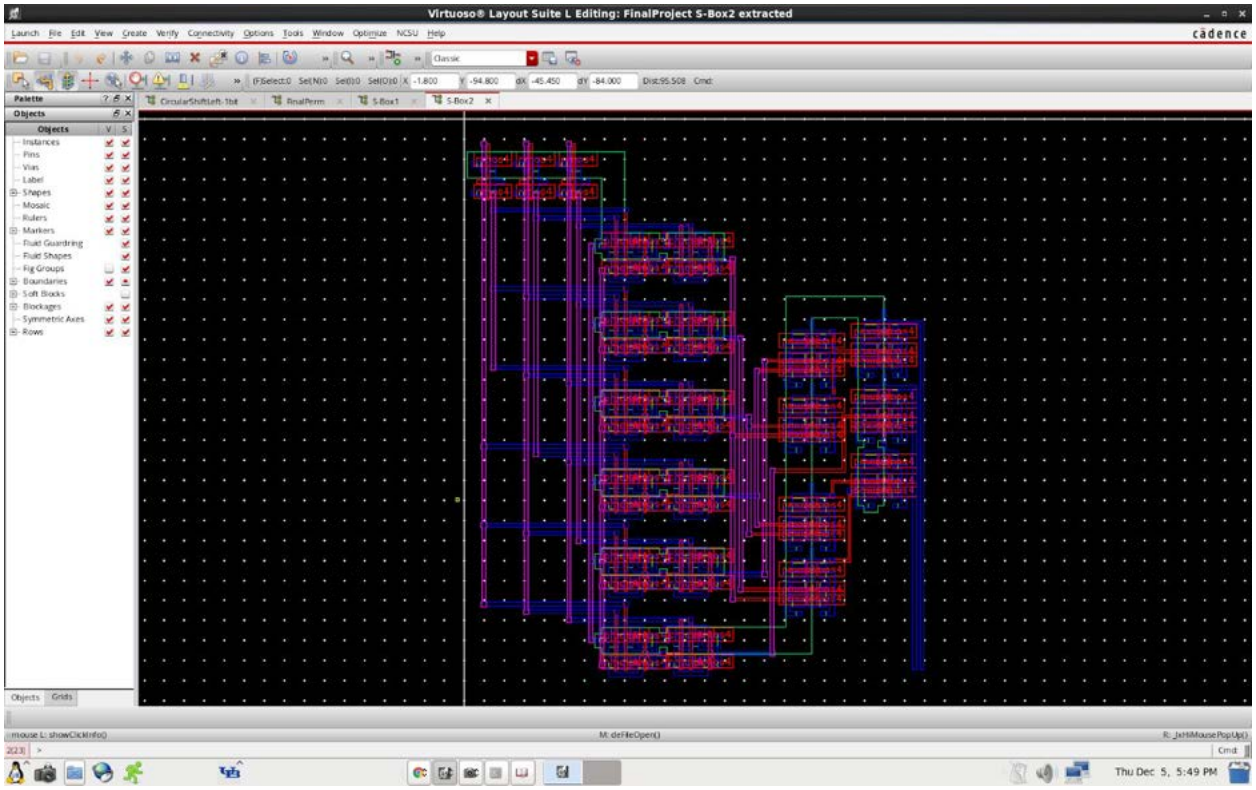


Figure 58: Extracted View of S-box 2

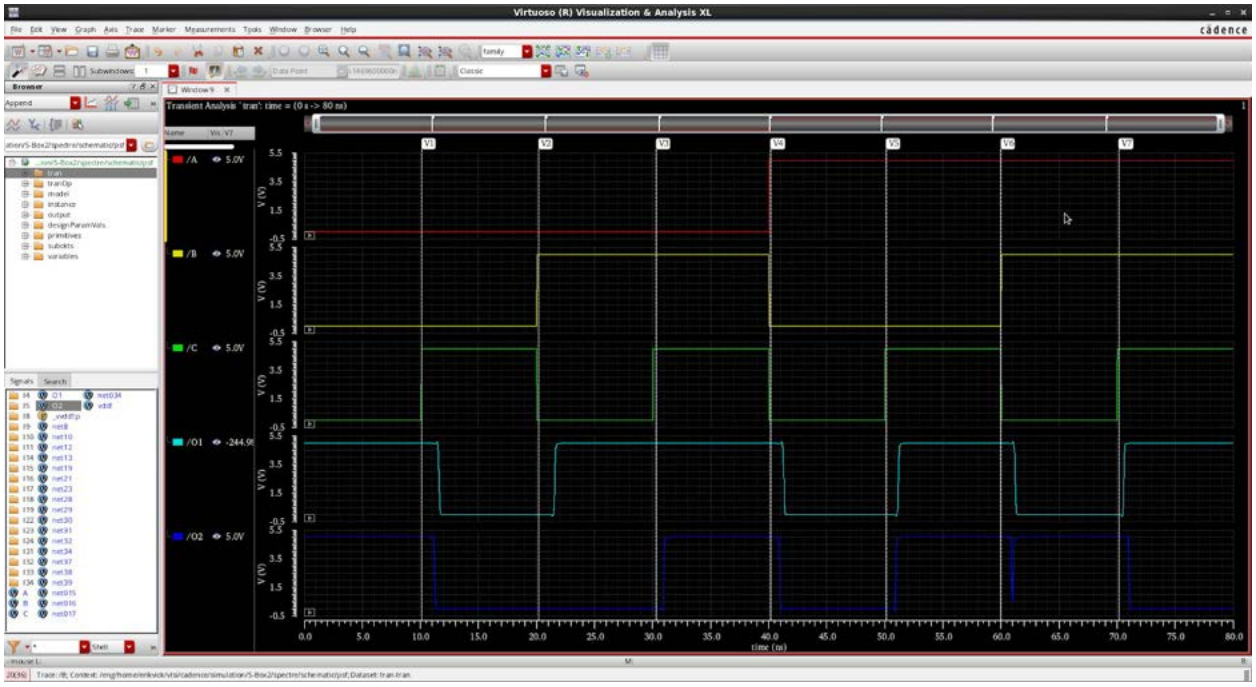


Figure 59: Schematic Simulation of S-box 2

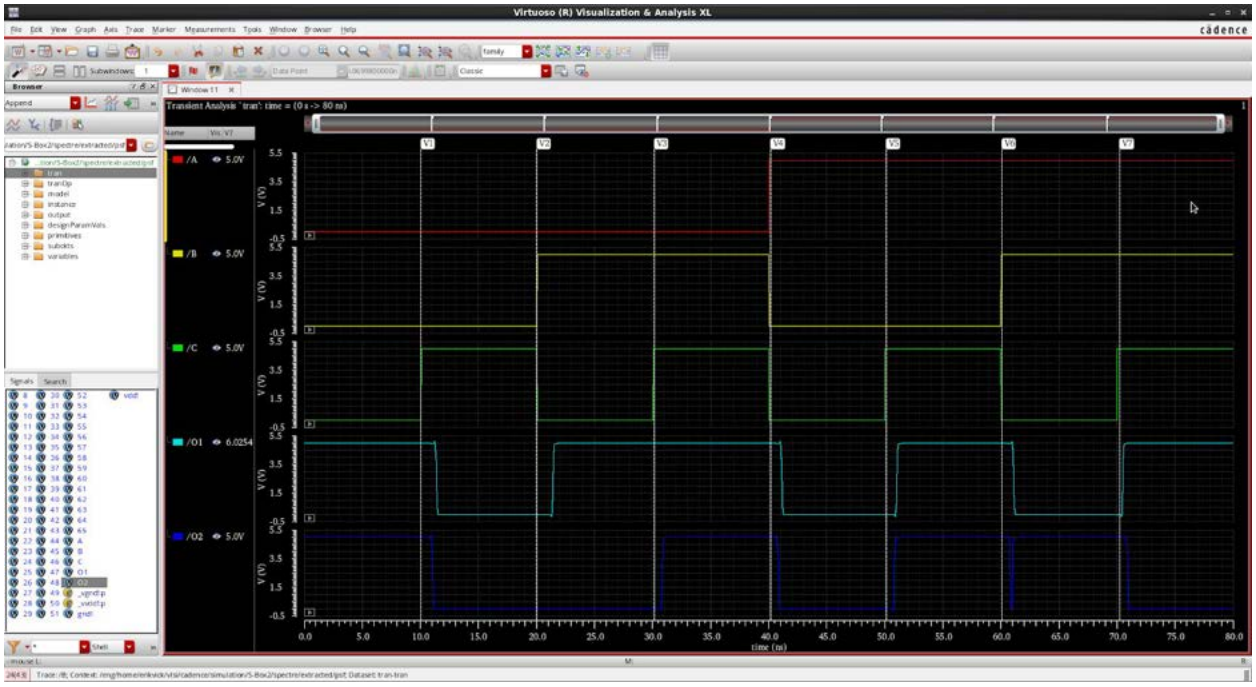


Figure 60: Extracted Simulation of S-box 2

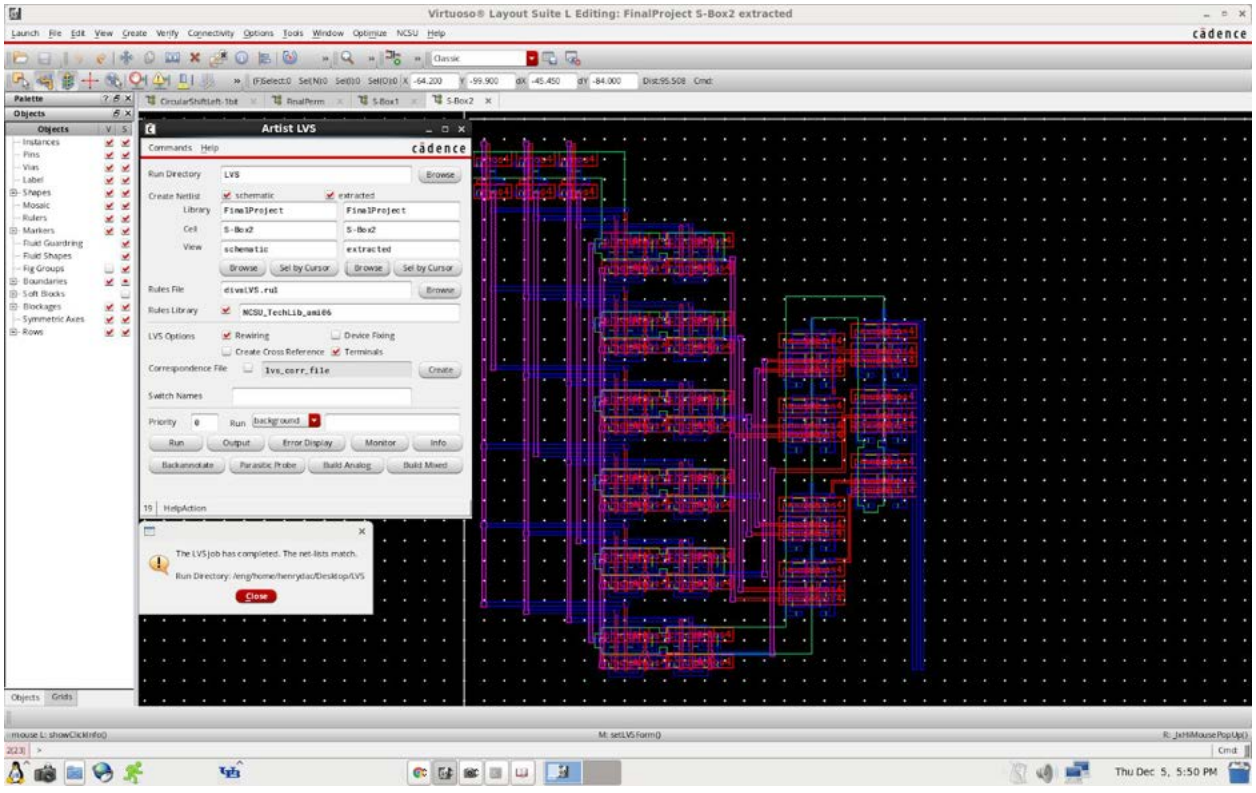


Figure 61: LVS Check for S-box 2

S-box 3 - Views:

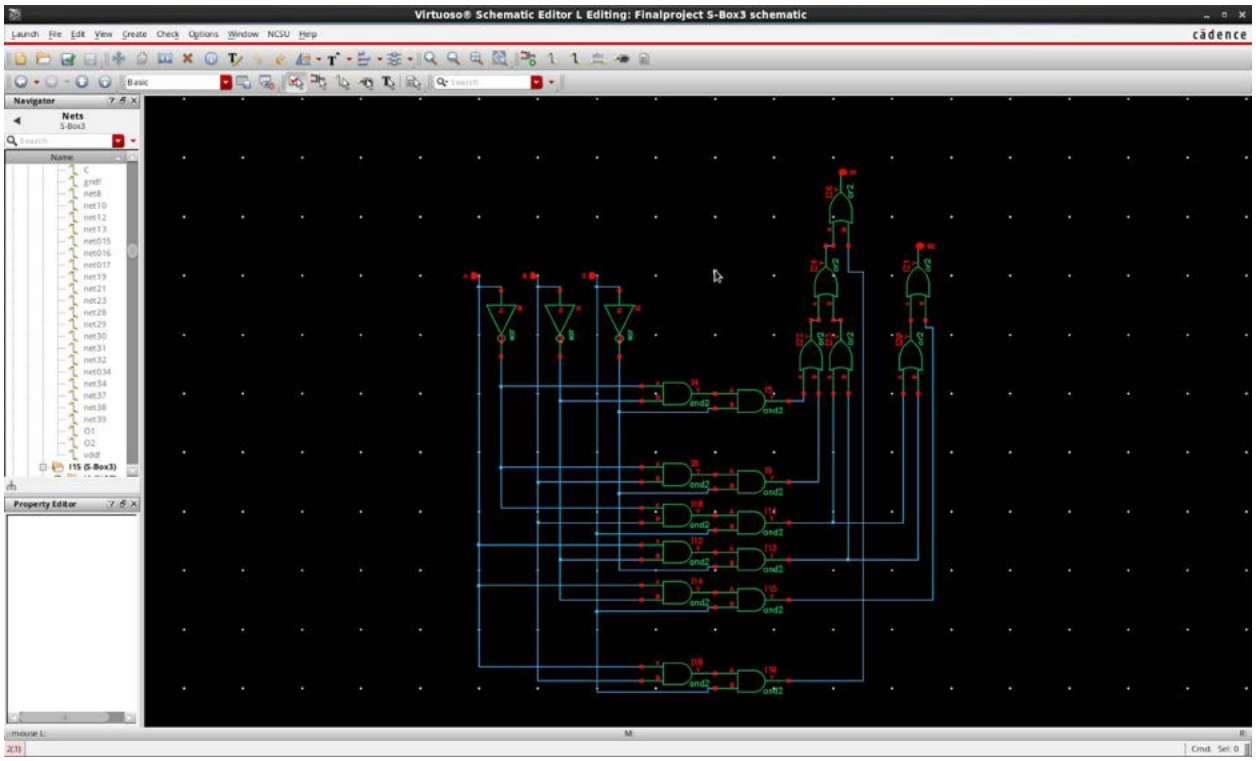


Figure 62: Schematic View of S-box 3

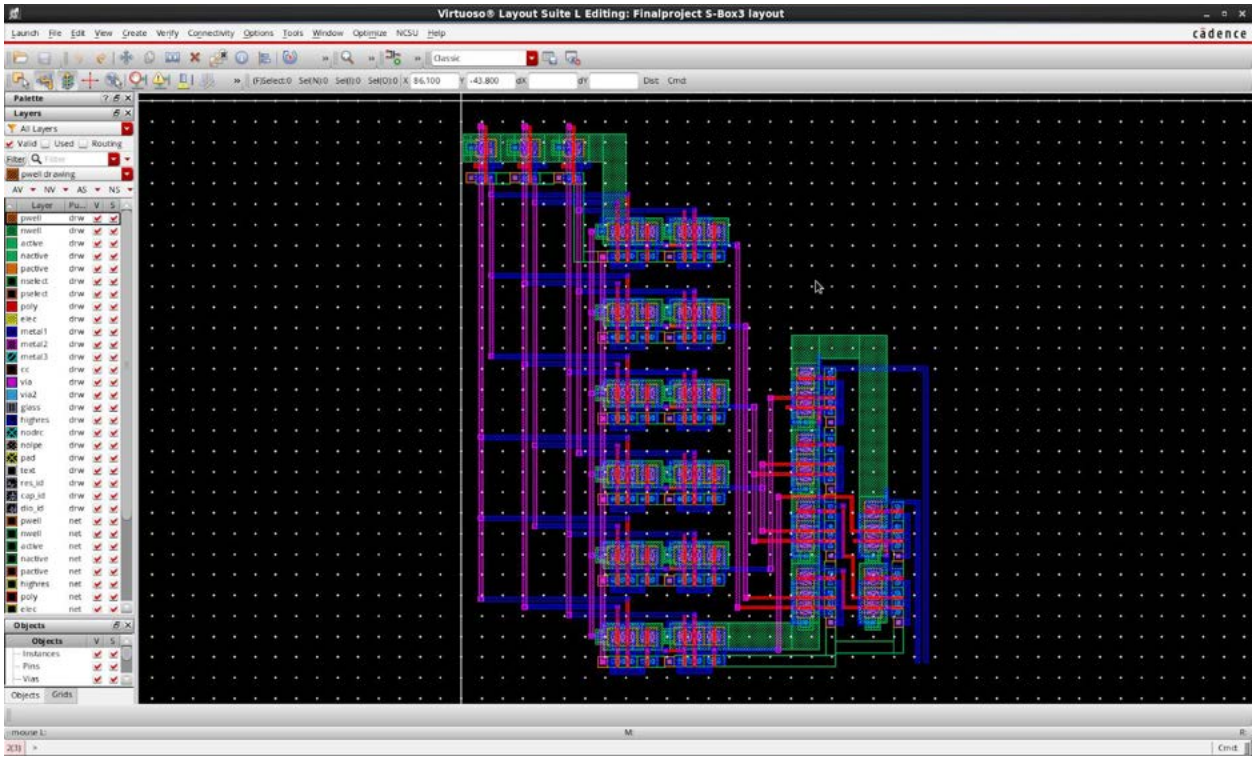


Figure 63: Layout View of S-box 3

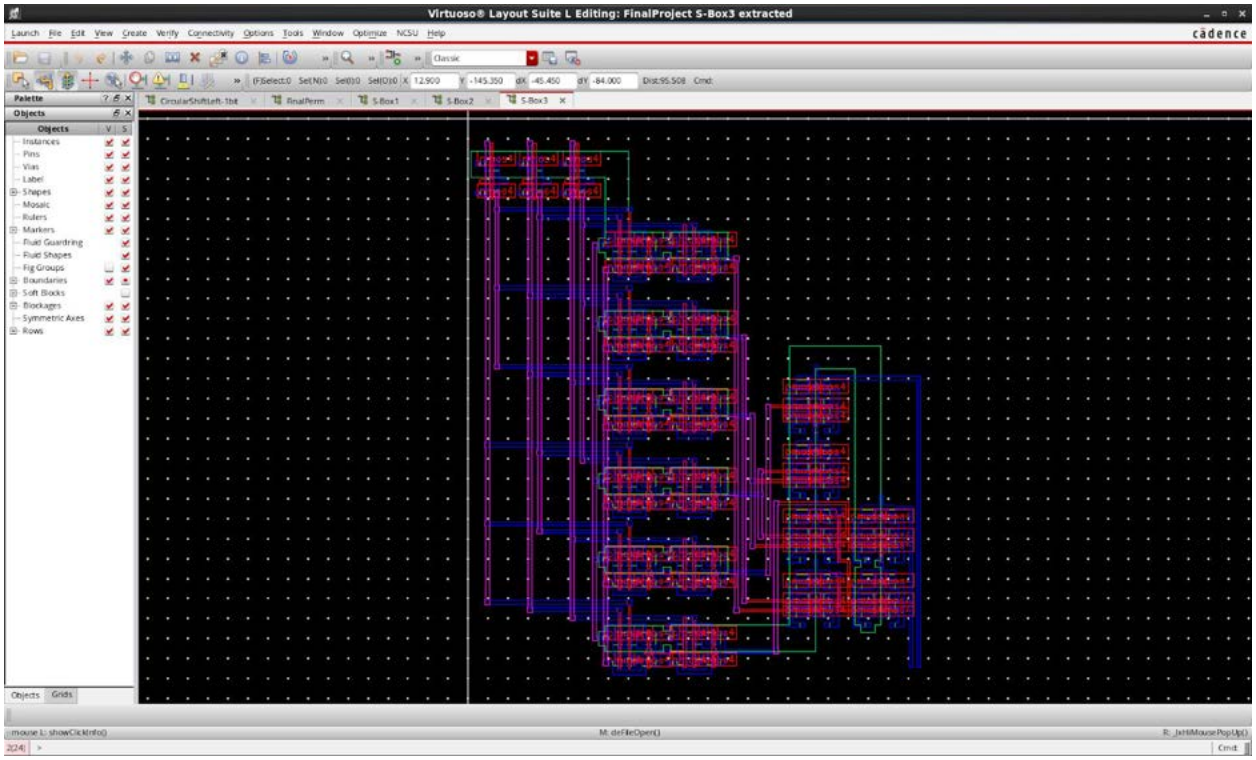


Figure 64: Extracted View of S-box 3

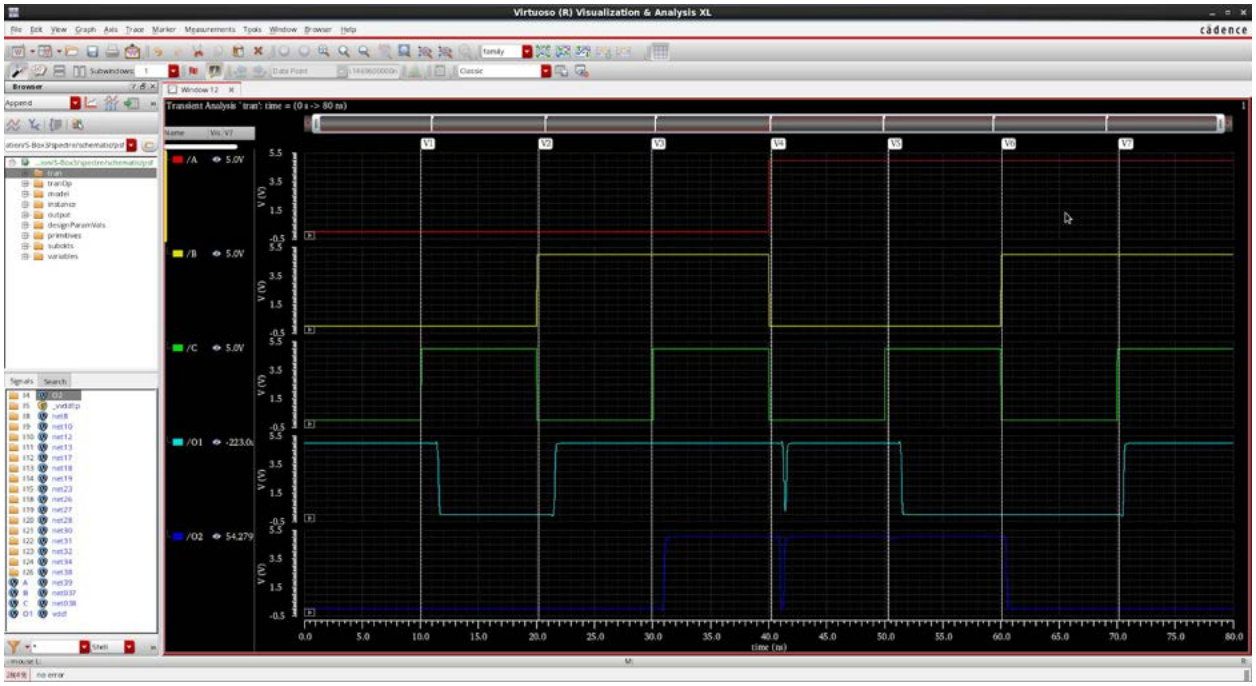


Figure 65: Schematic Simulation of S-box 3

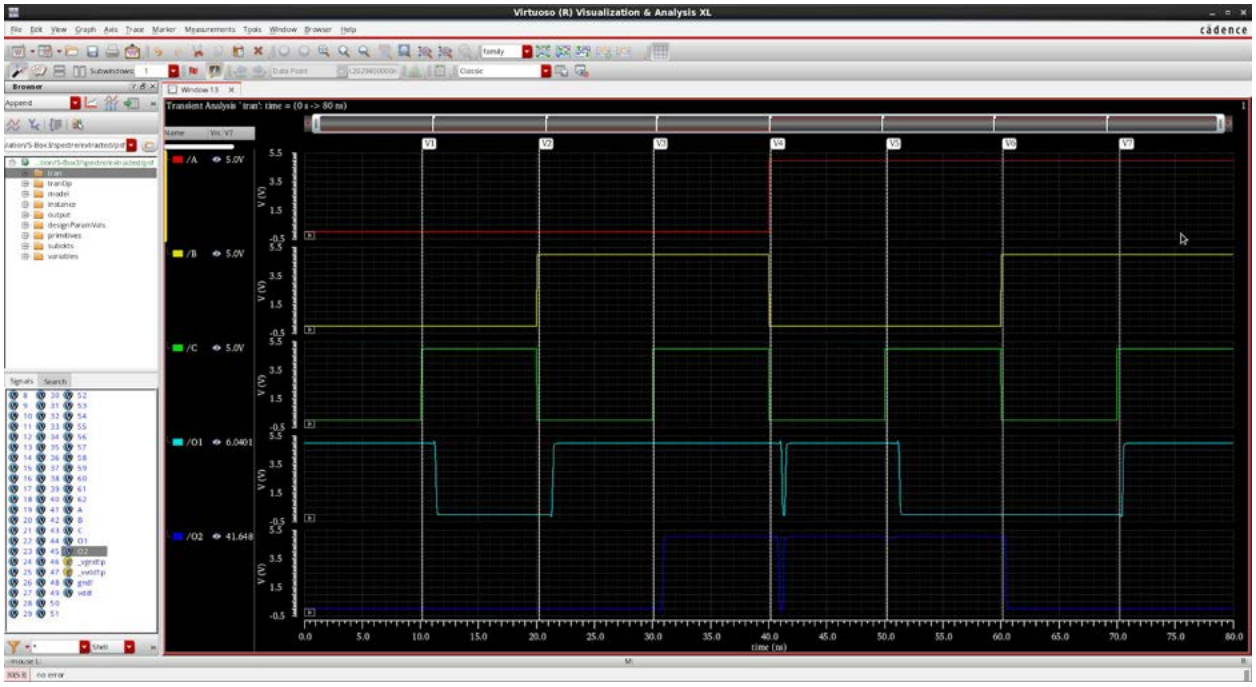


Figure 66: Extracted Simulation of S-box 3

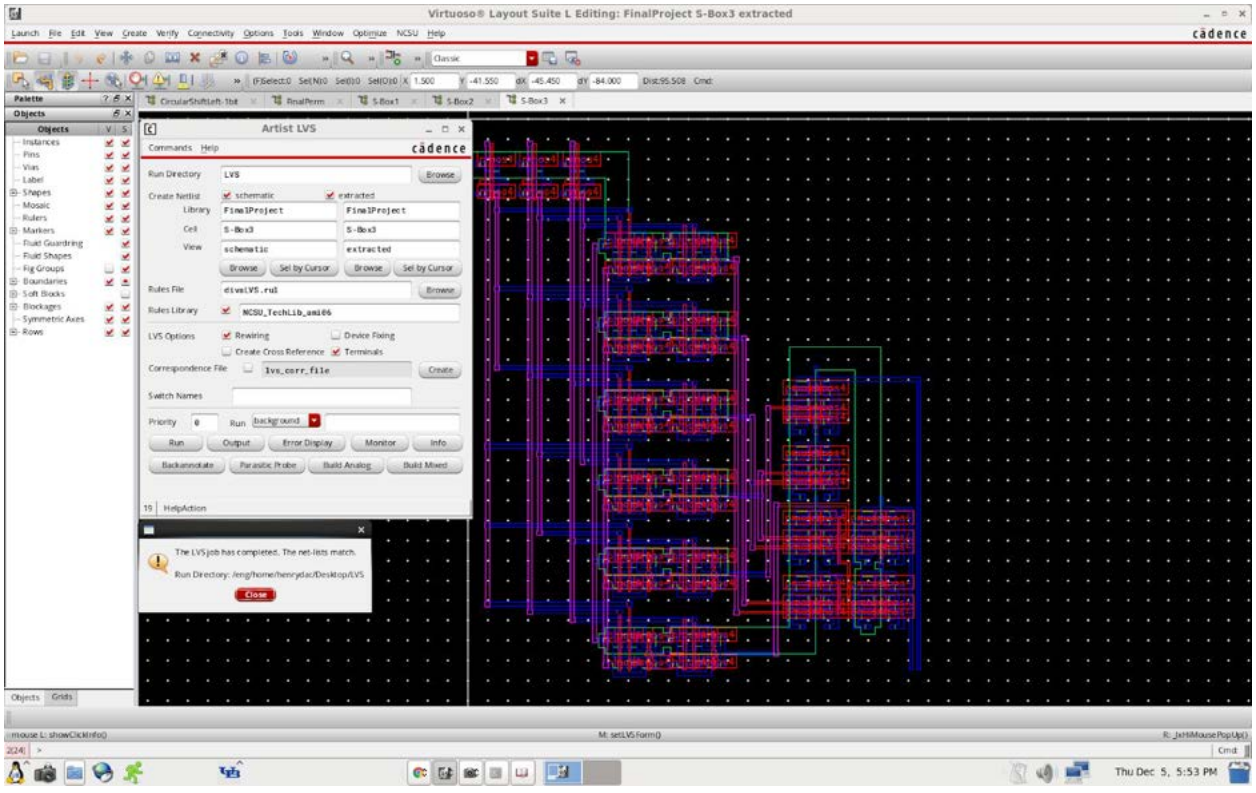


Figure 67: LVS Check for S-box 3

S-box 4 - Views:

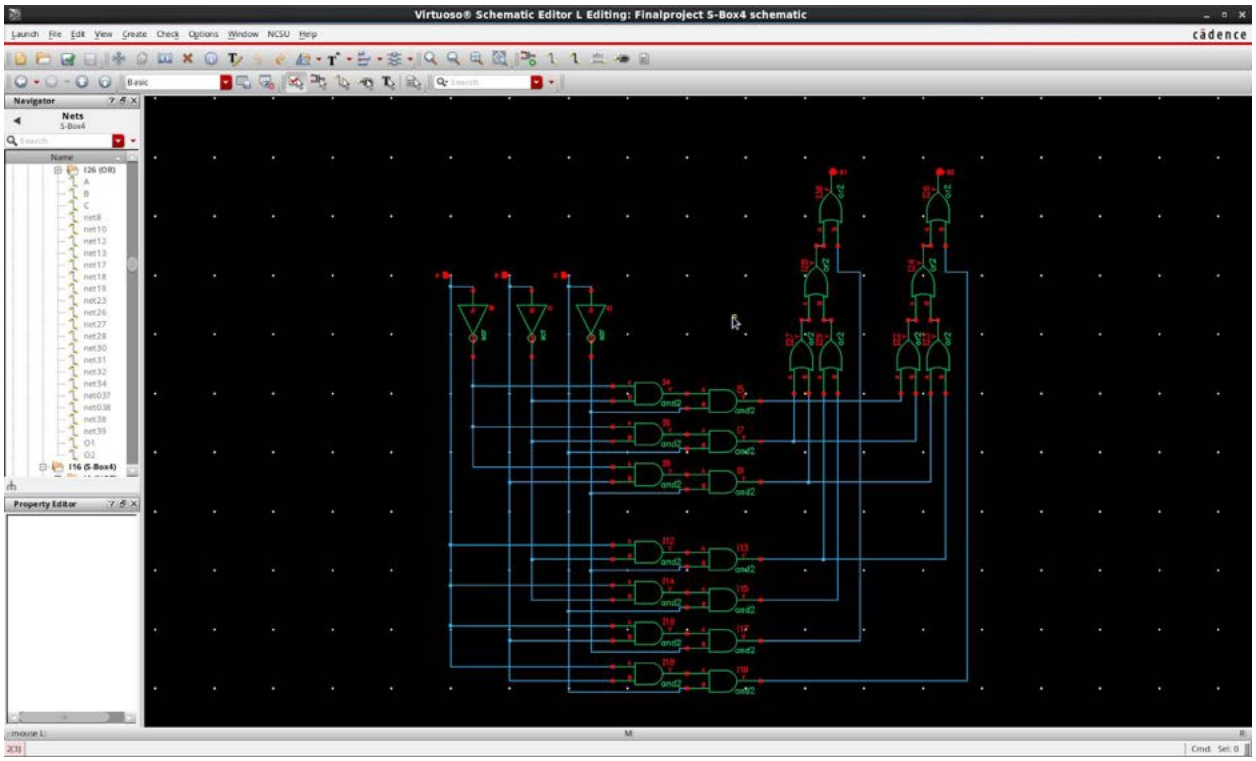


Figure 68: Schematic View of S-box 4

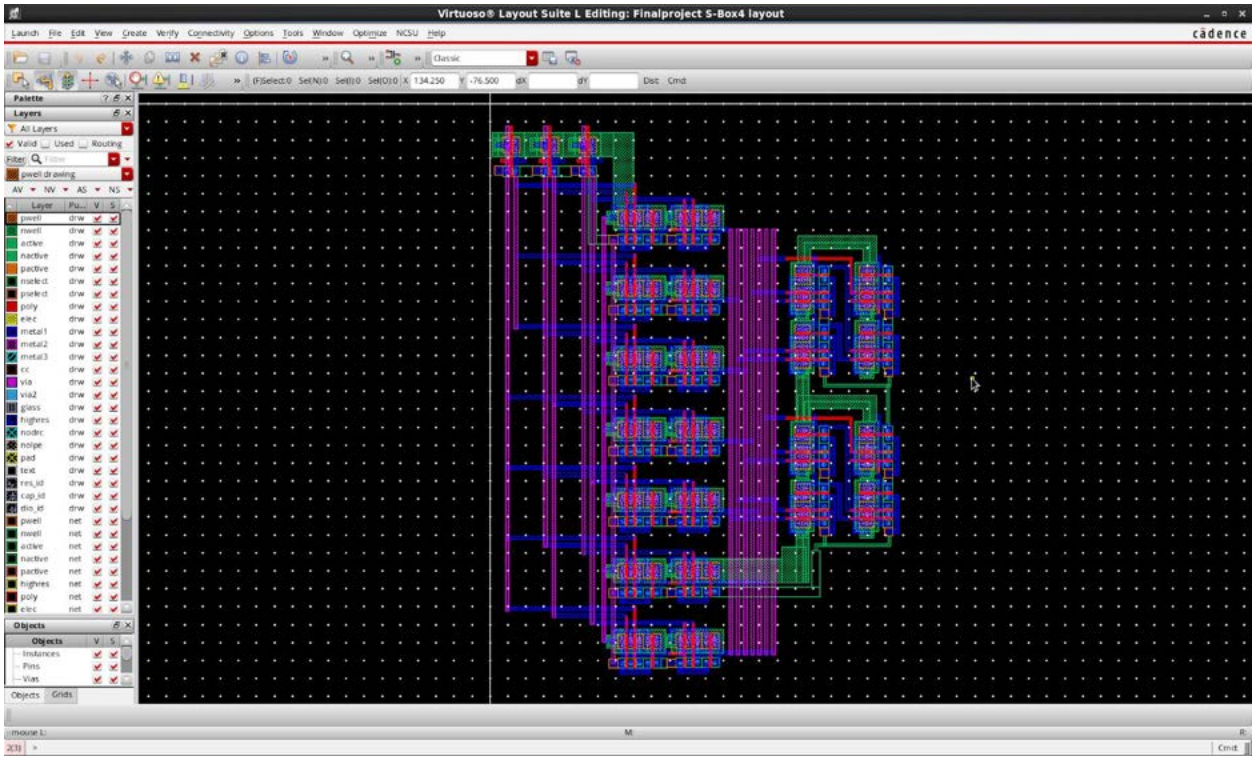


Figure 69: Layout View of S-box 4

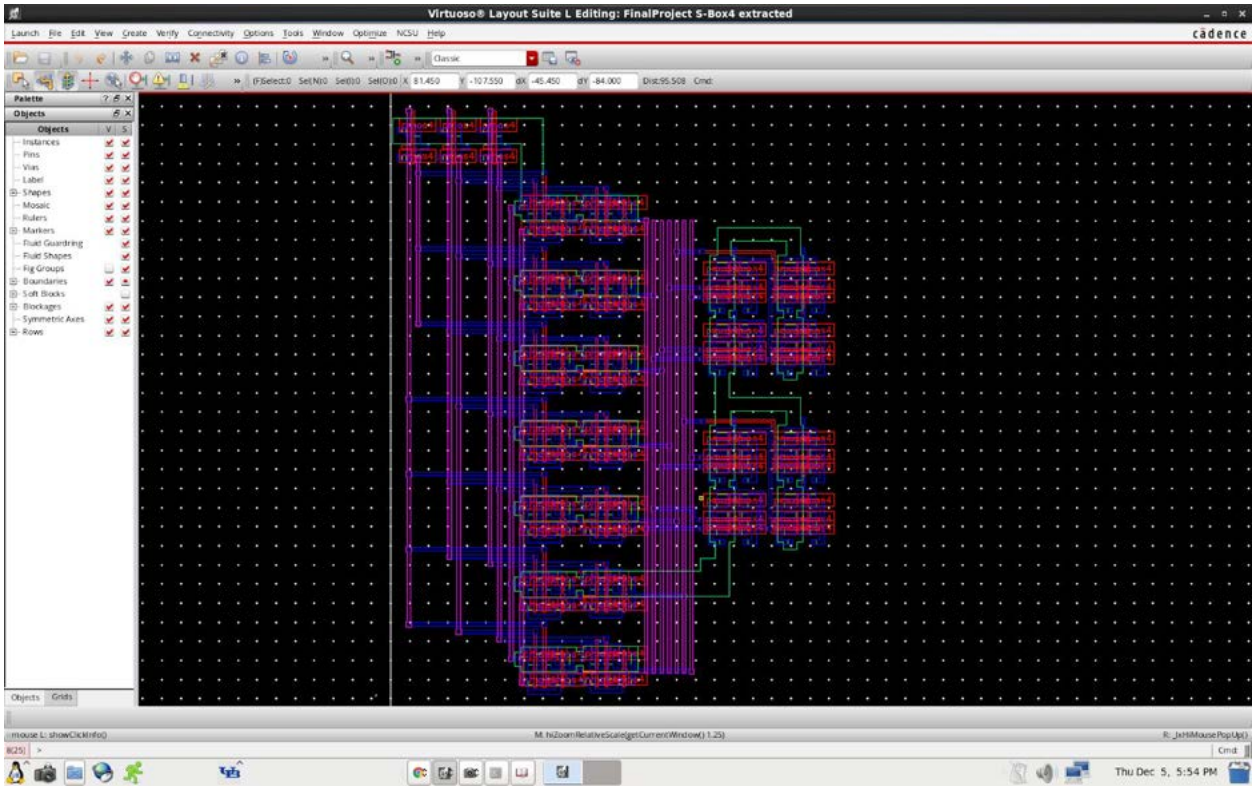


Figure 70: Extracted View of S-box 4

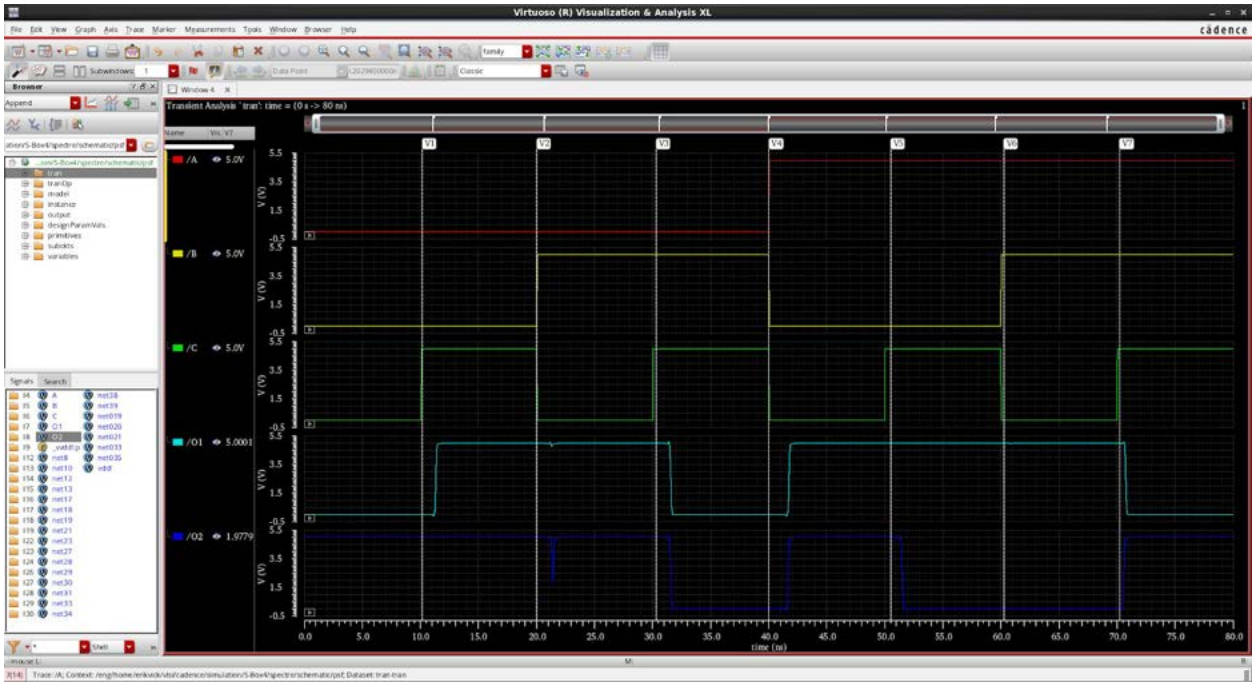


Figure 71: Schematic Simulation of S-box 4

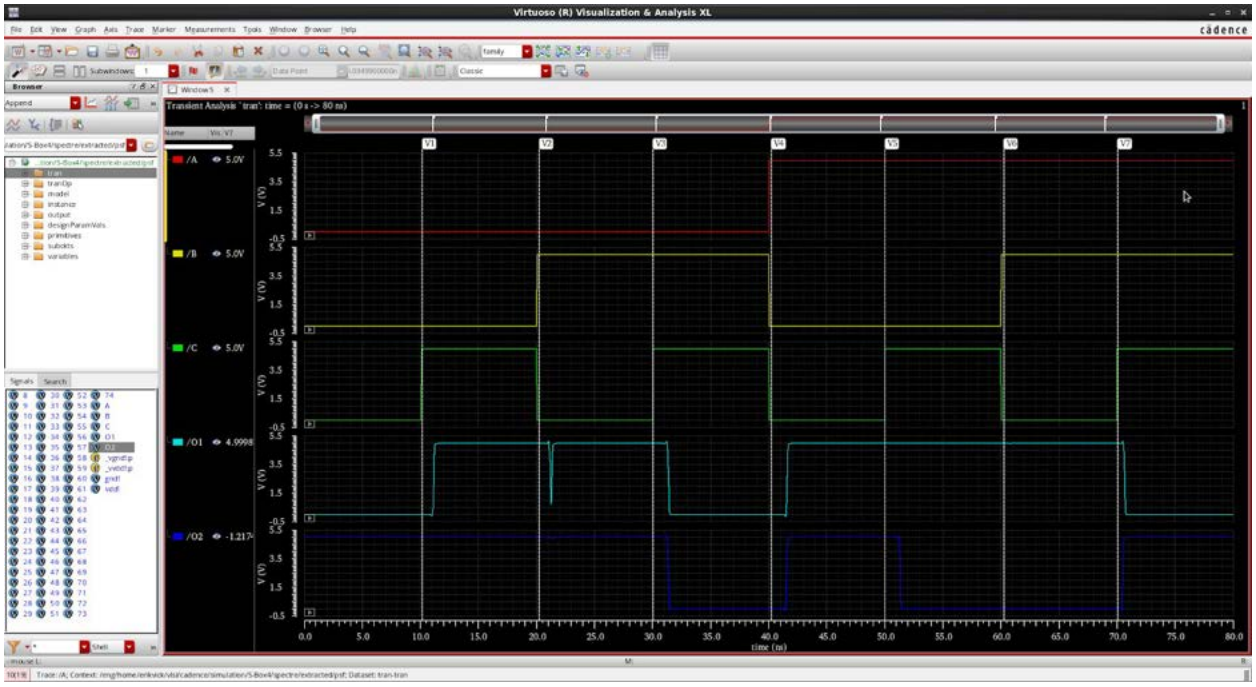


Figure 72: Extracted Simulation of S-box 4

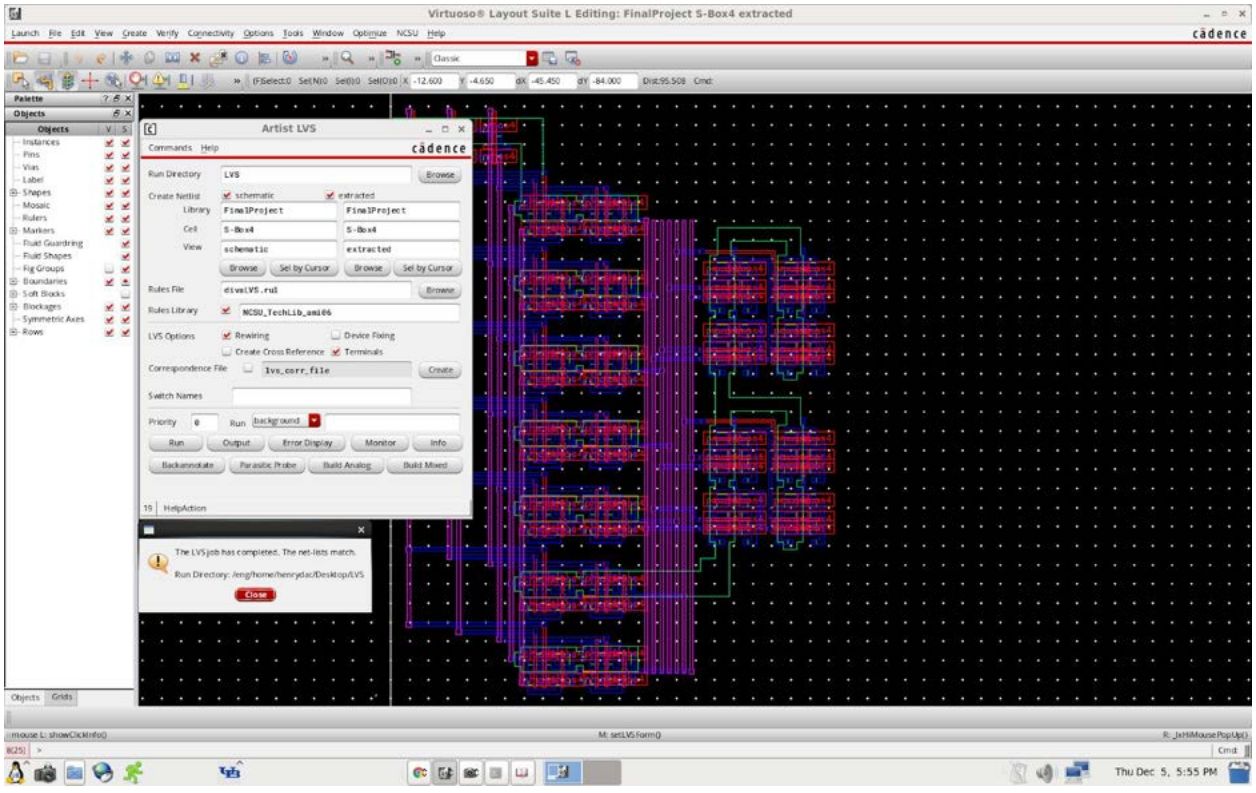


Figure 73: LVS Check for S-box 4

Subkey Generation 1 - Views:

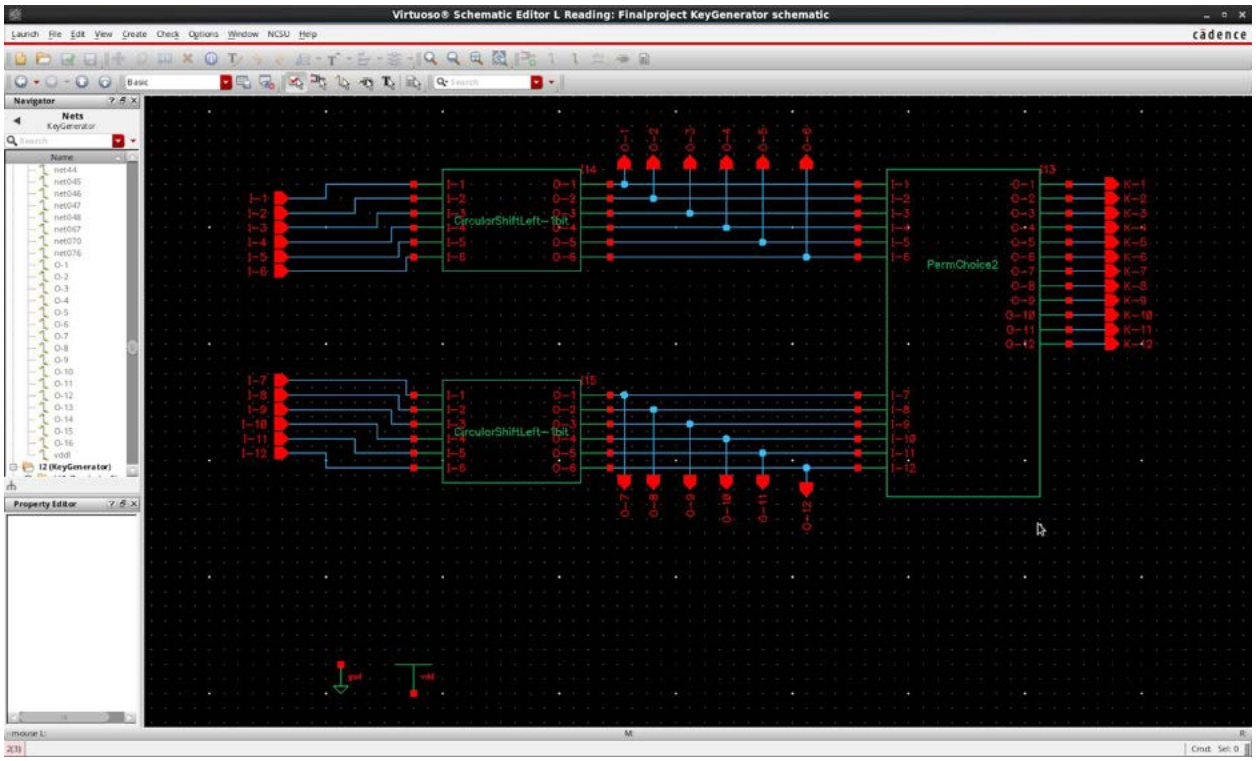


Figure 74: Schematic View of Subkey Generation 1

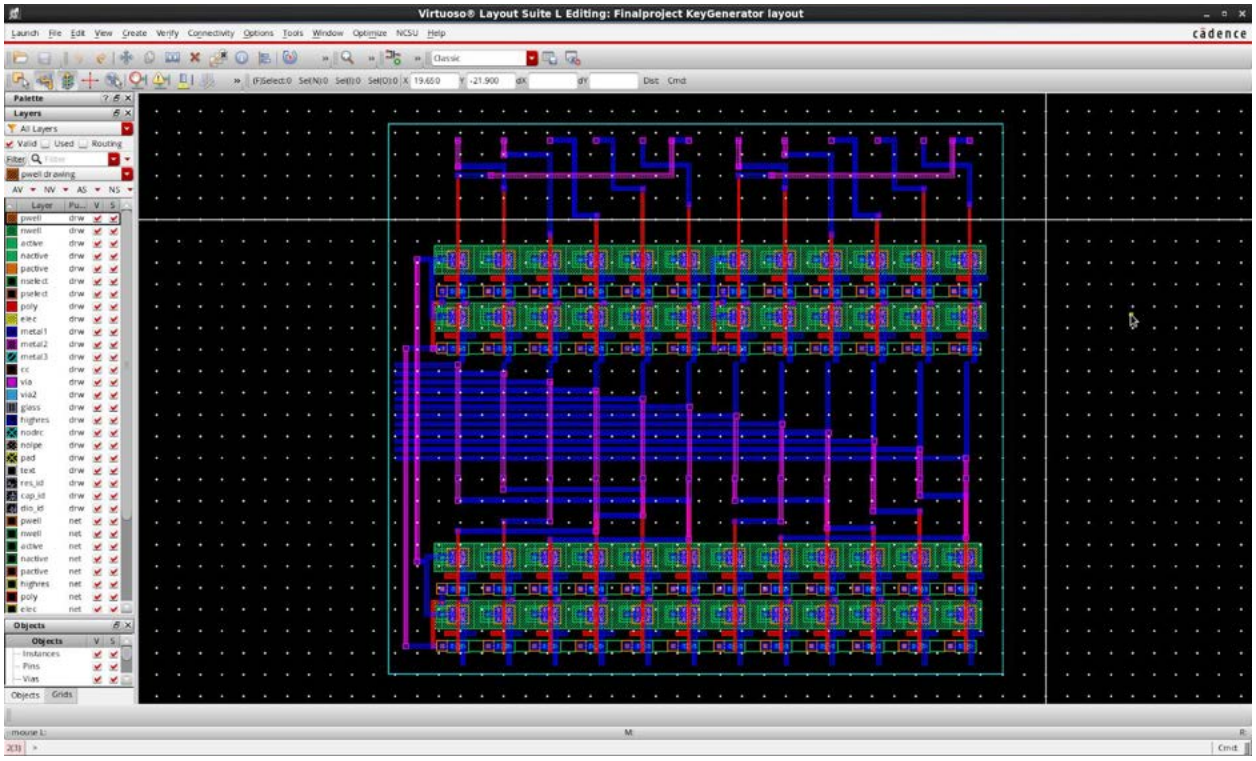


Figure 75: Layout View of Subkey Generation 1

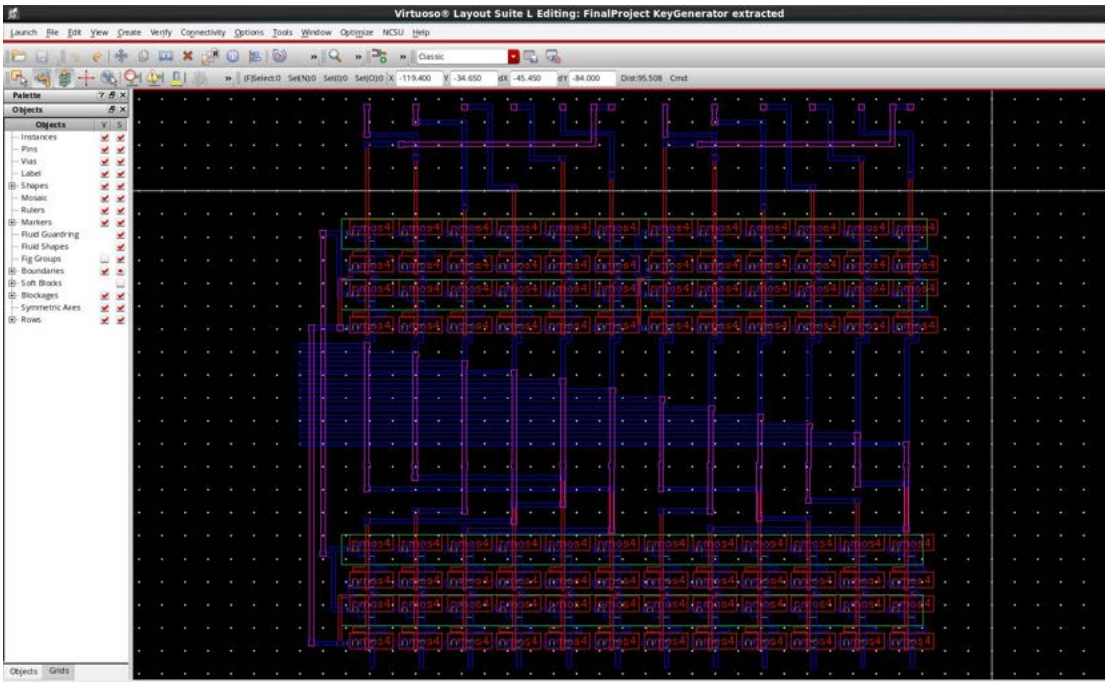


Figure 76: Extracted View of Subkey Generation 1

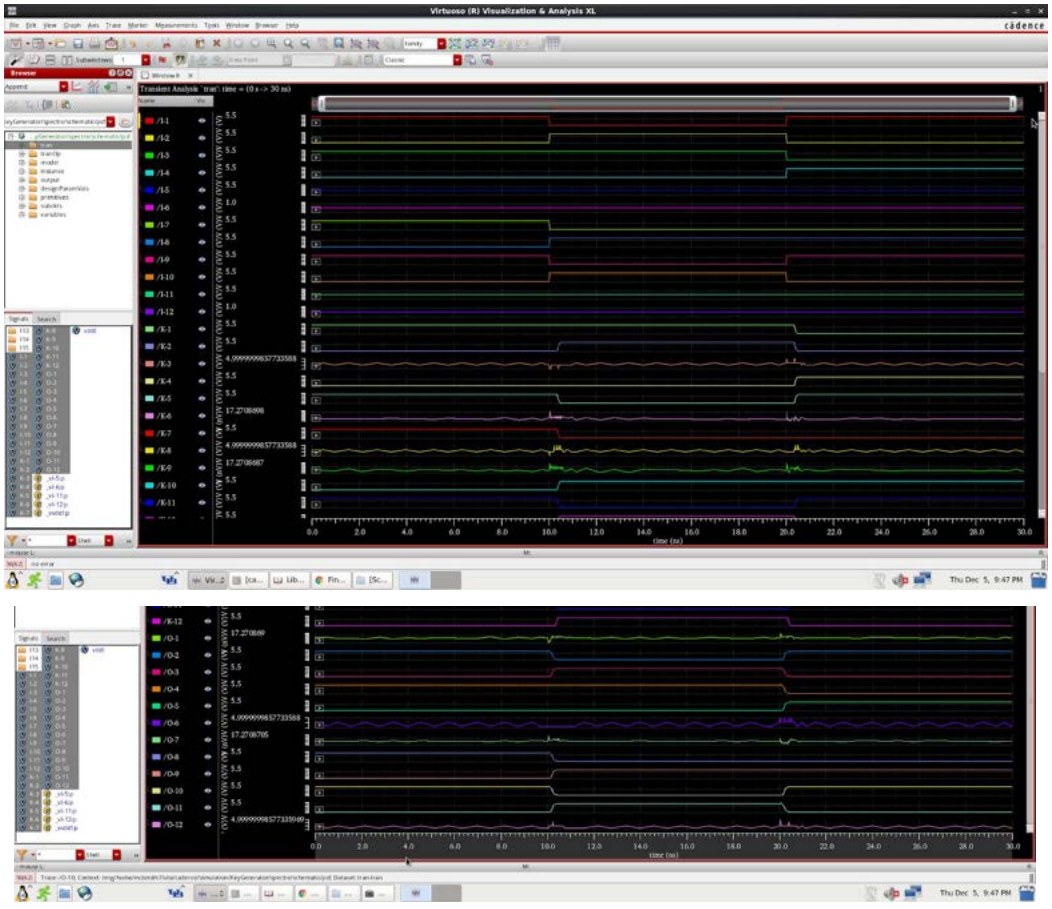


Figure 77: Schematic Simulation of Subkey Generation 1

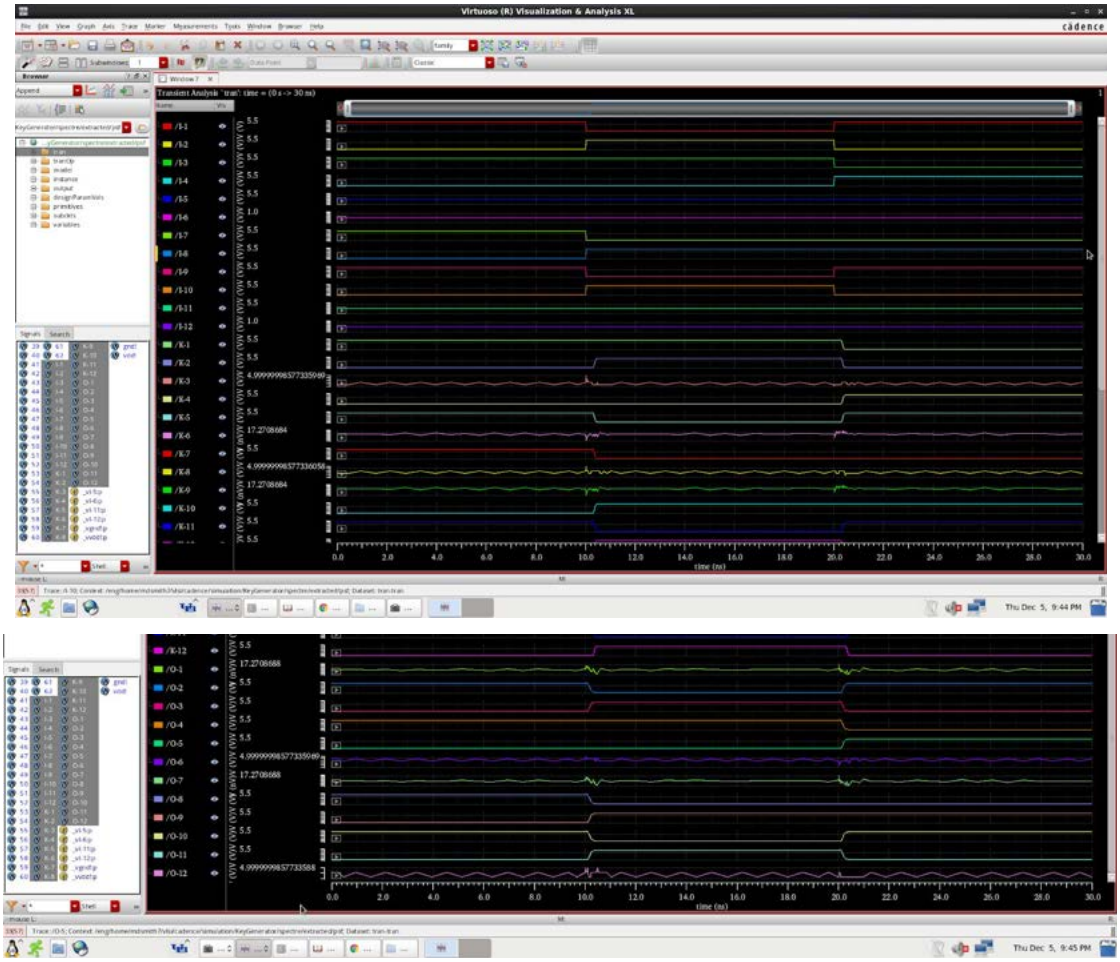


Figure 78: Extracted Simulation of Subkey Generation 1

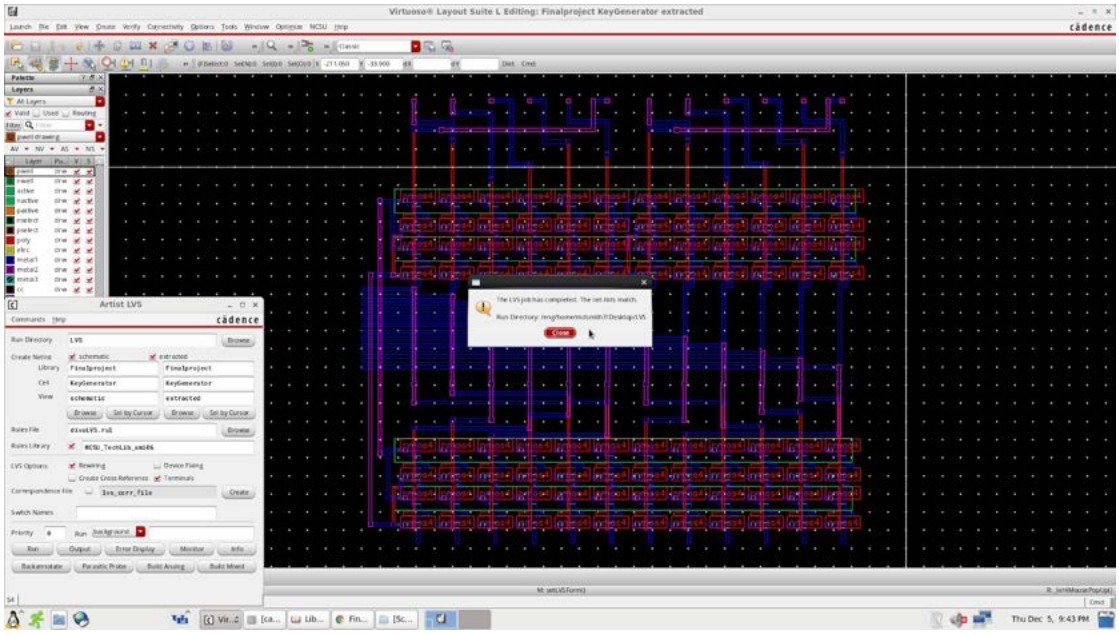


Figure 79: LVS Check for Subkey Generation 1

Subkey Generation 2 - Views:

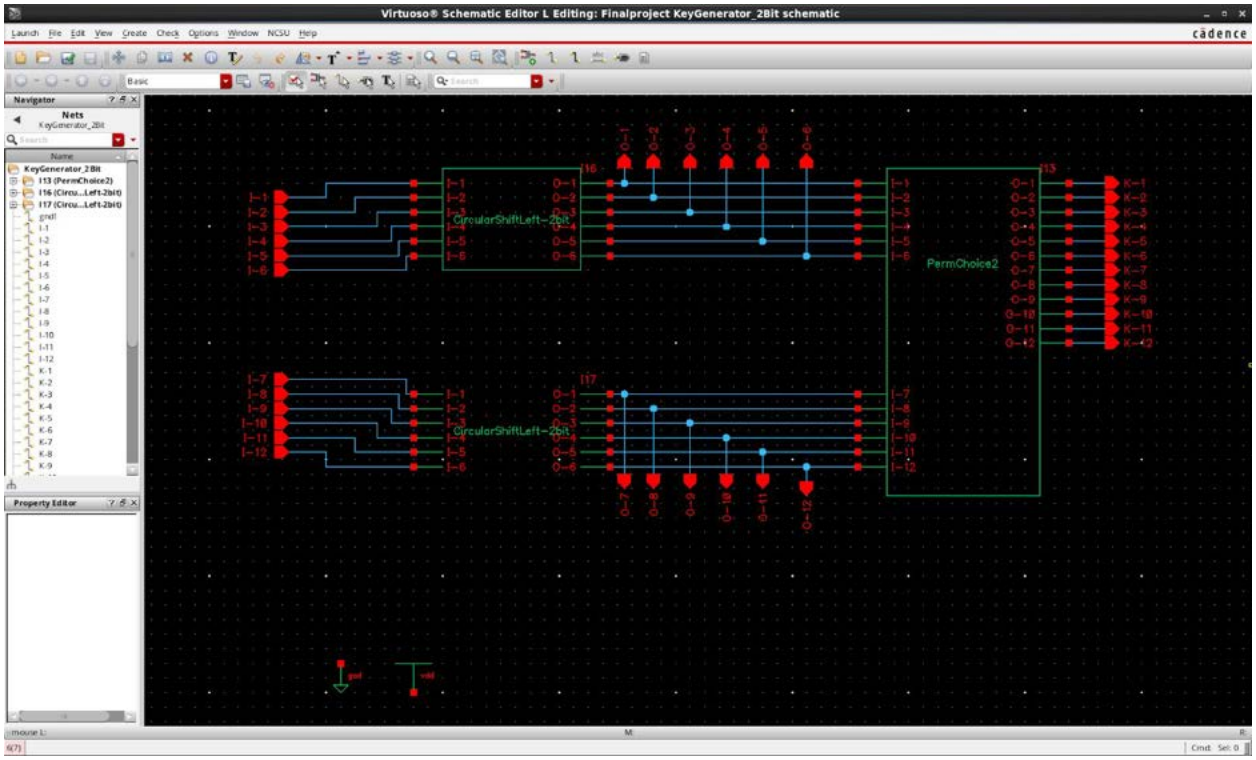


Figure 80: Schematic View of Subkey Generation 2

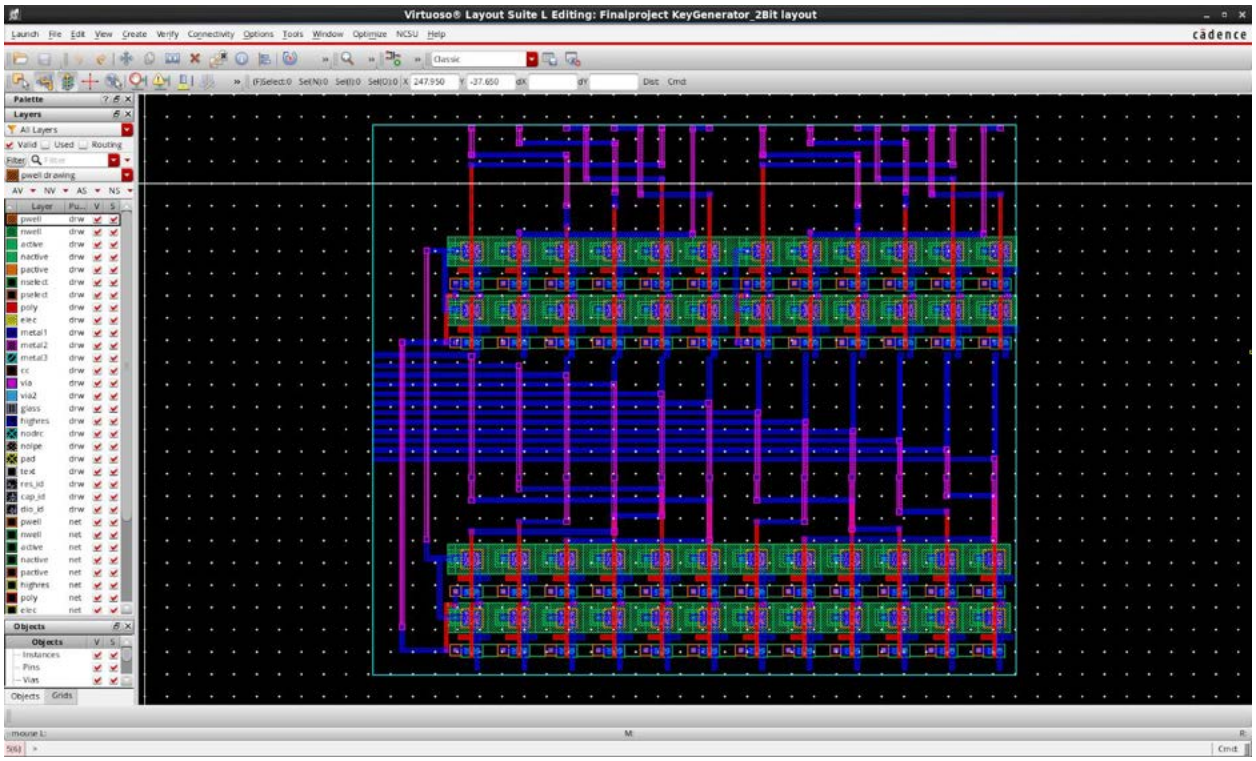


Figure 81: Layout View of Subkey Generation 2

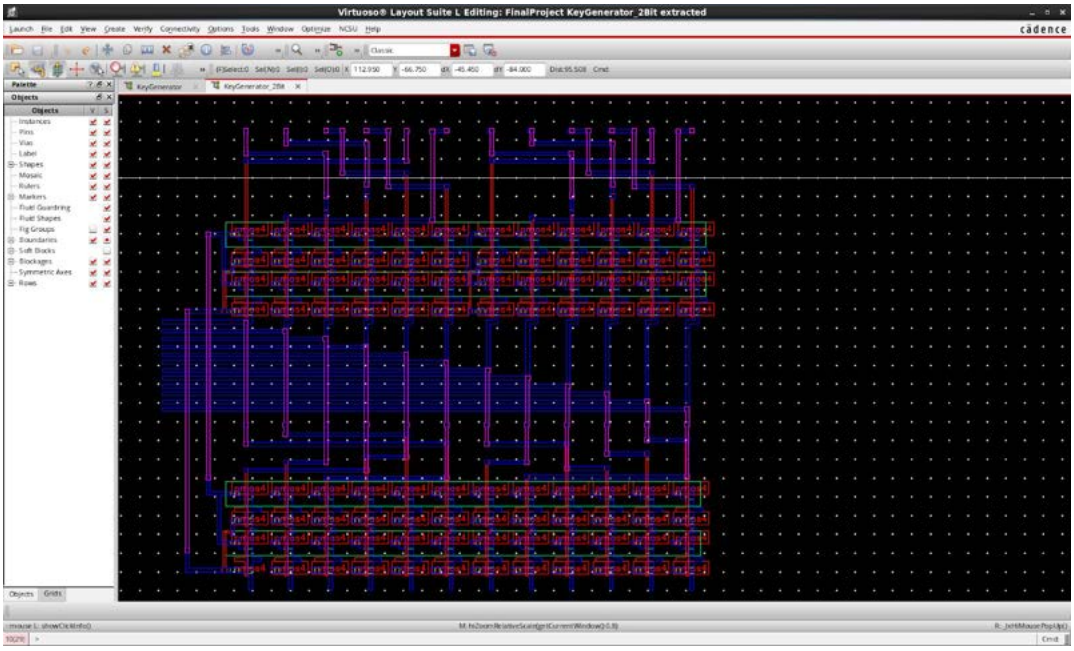


Figure 82: Extracted View of Subkey Generation 2

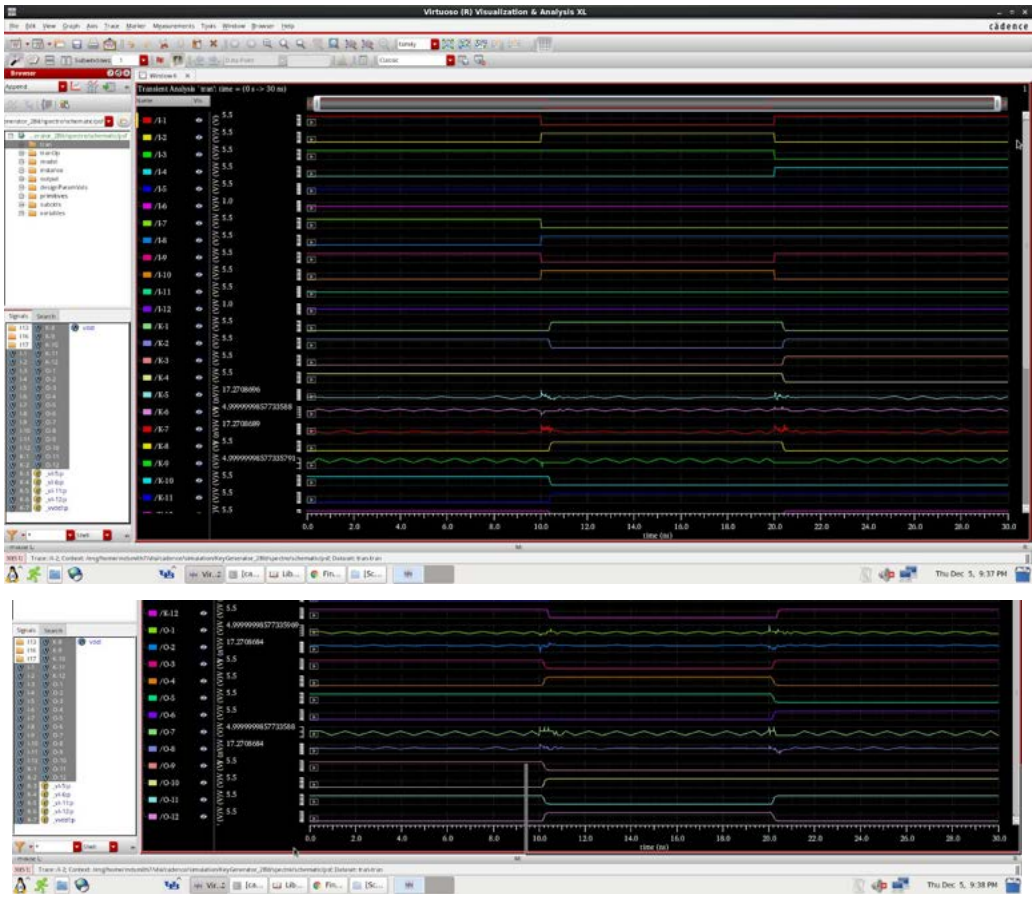


Figure 83: Schematic Simulation of Subkey Generation 2

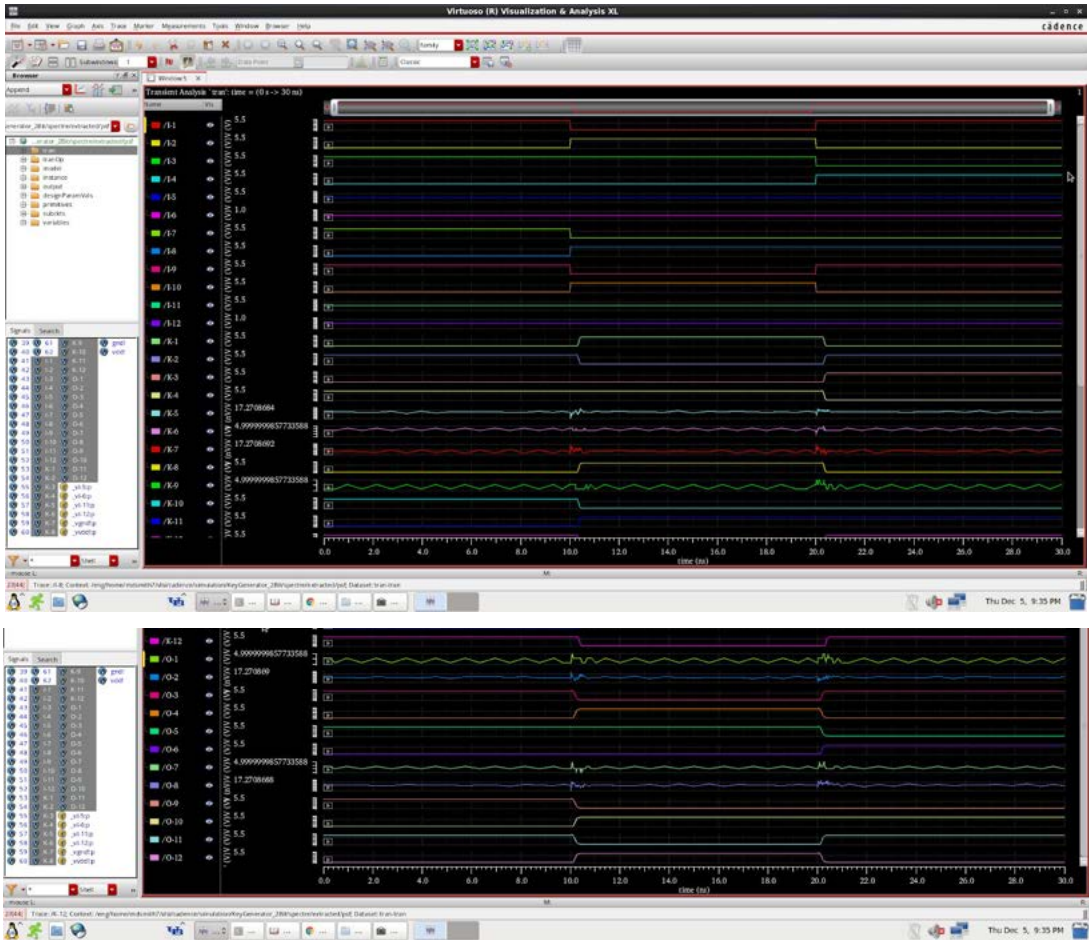


Figure 84: Extracted Simulation of Subkey Generation 2

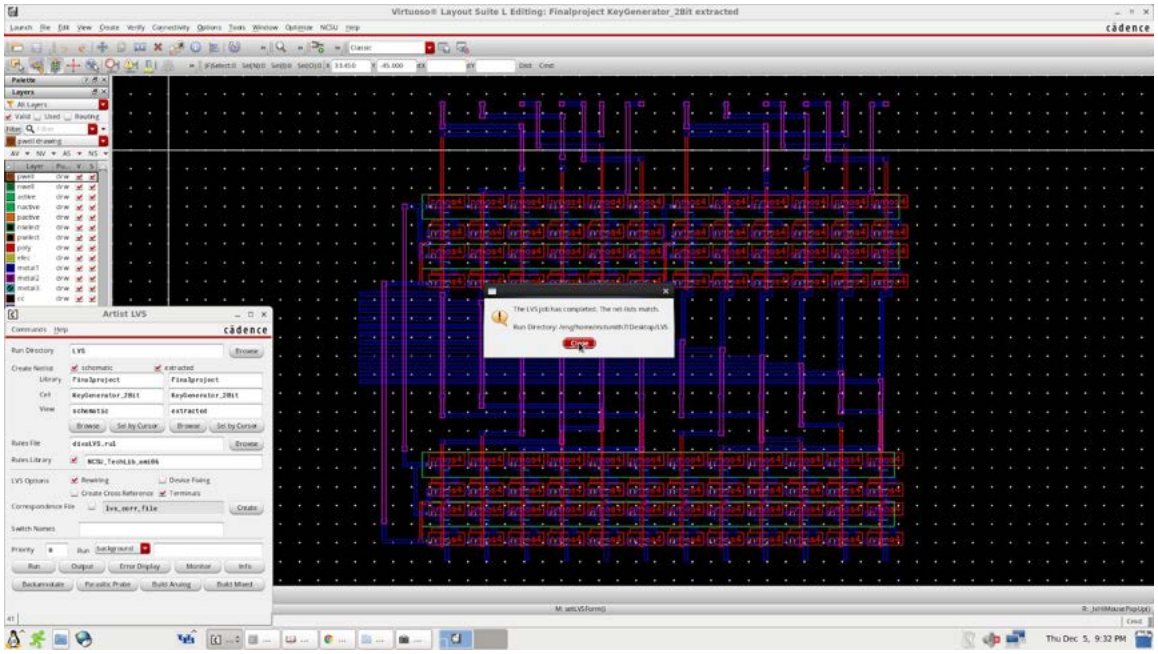


Figure 85: LVS Check for Subkey Generation 2

Cipher Function - Views:

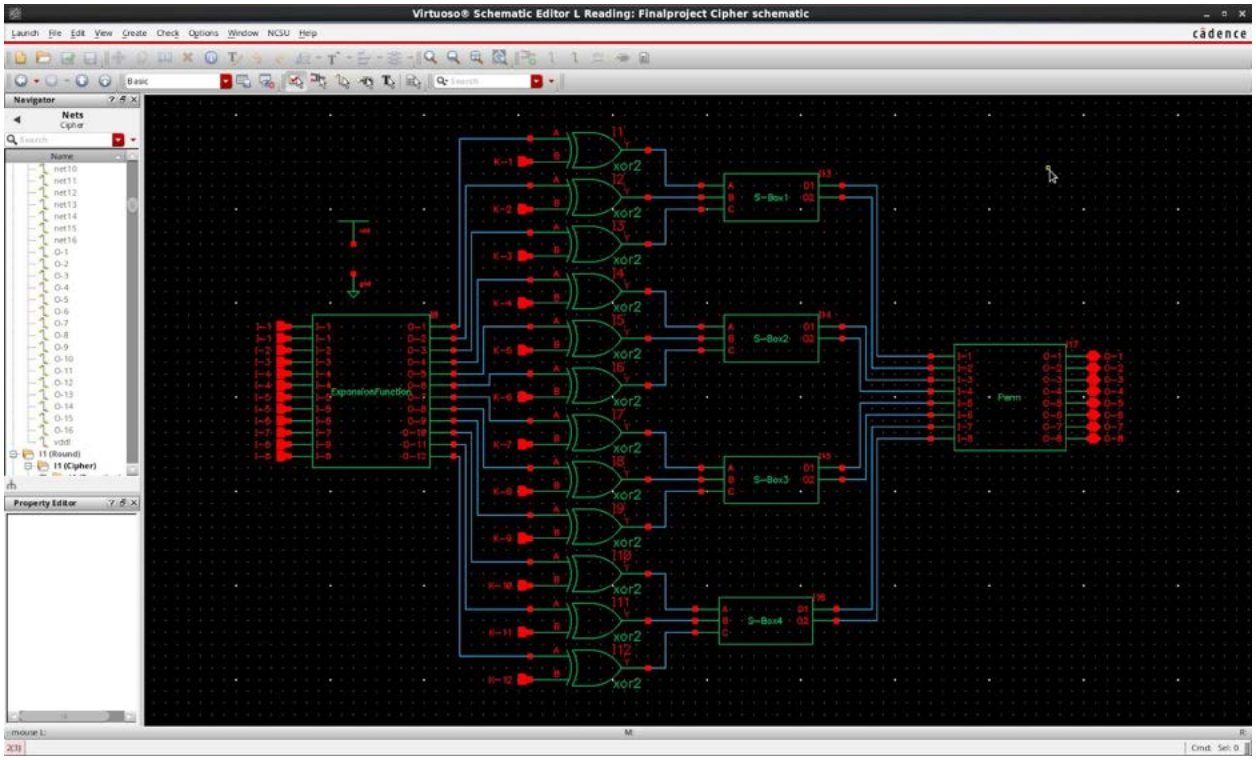


Figure 86: Schematic View of the Cipher Function

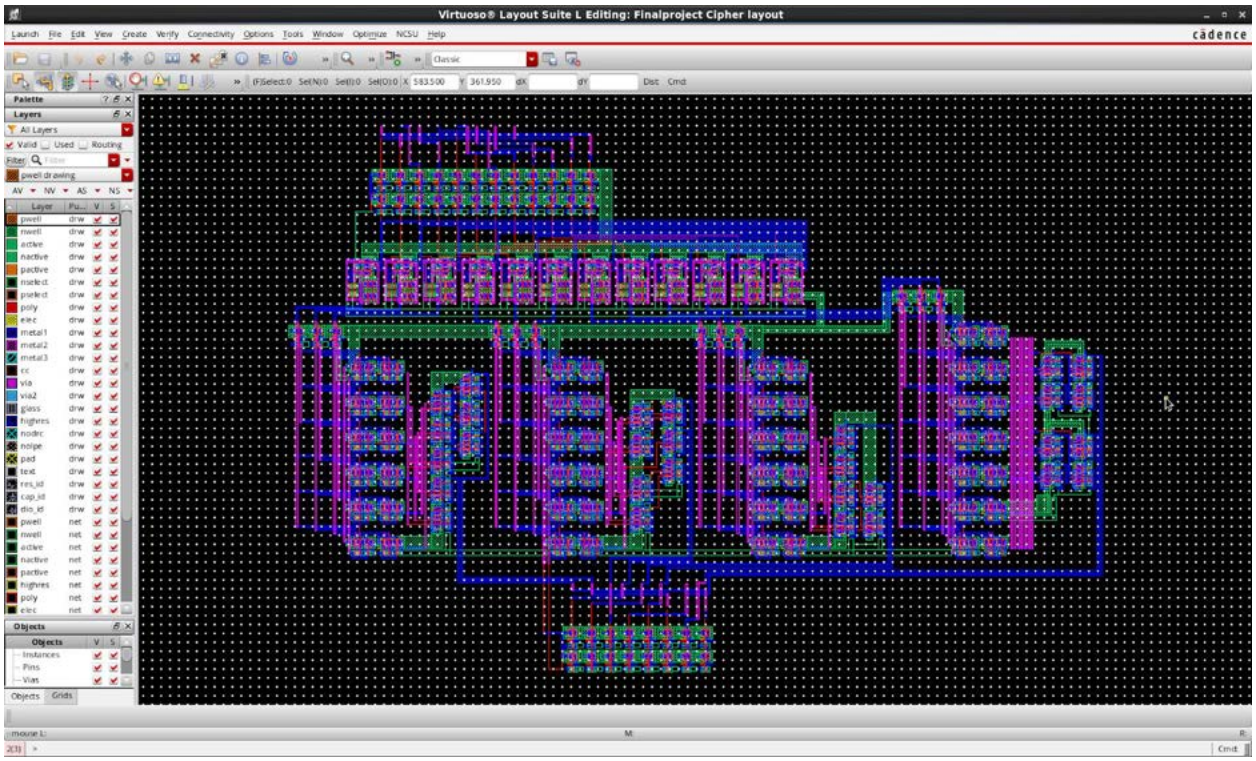


Figure 87: Layout View of the Cipher Function

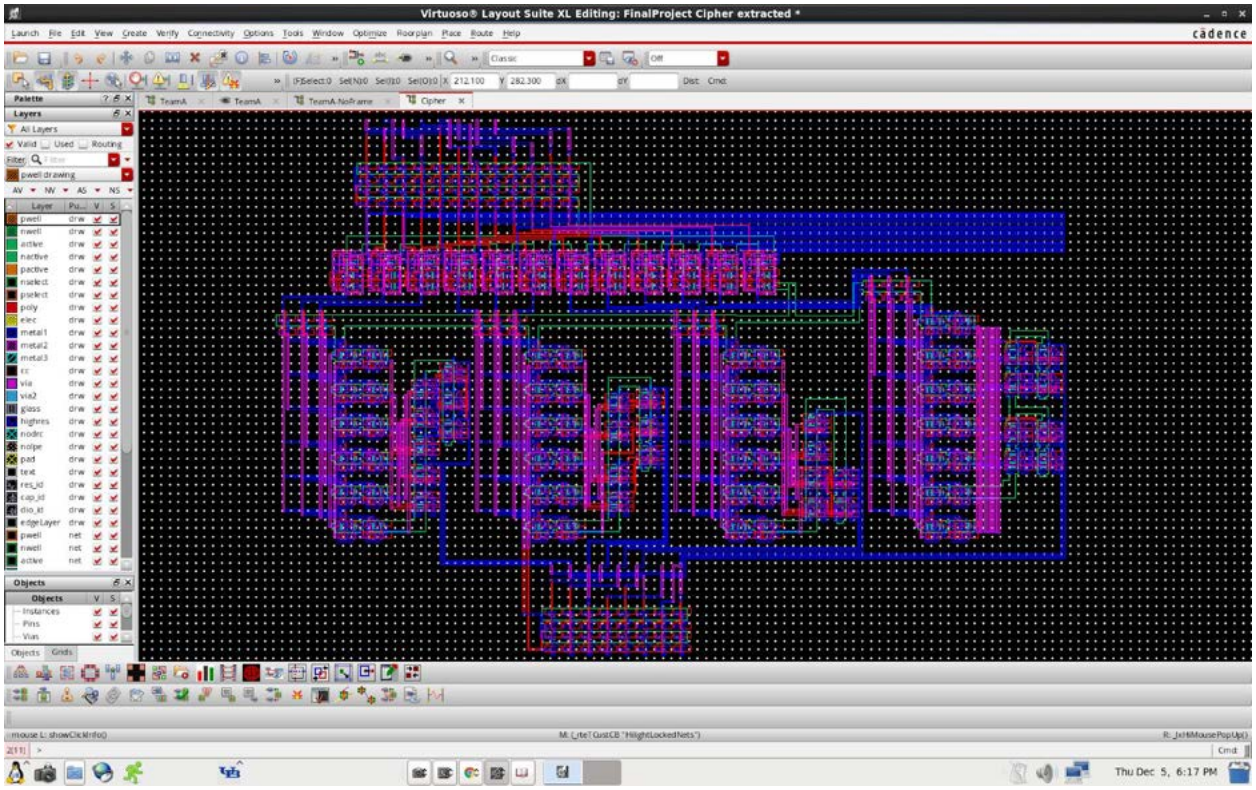


Figure 88: Extracted View of the Cipher Function

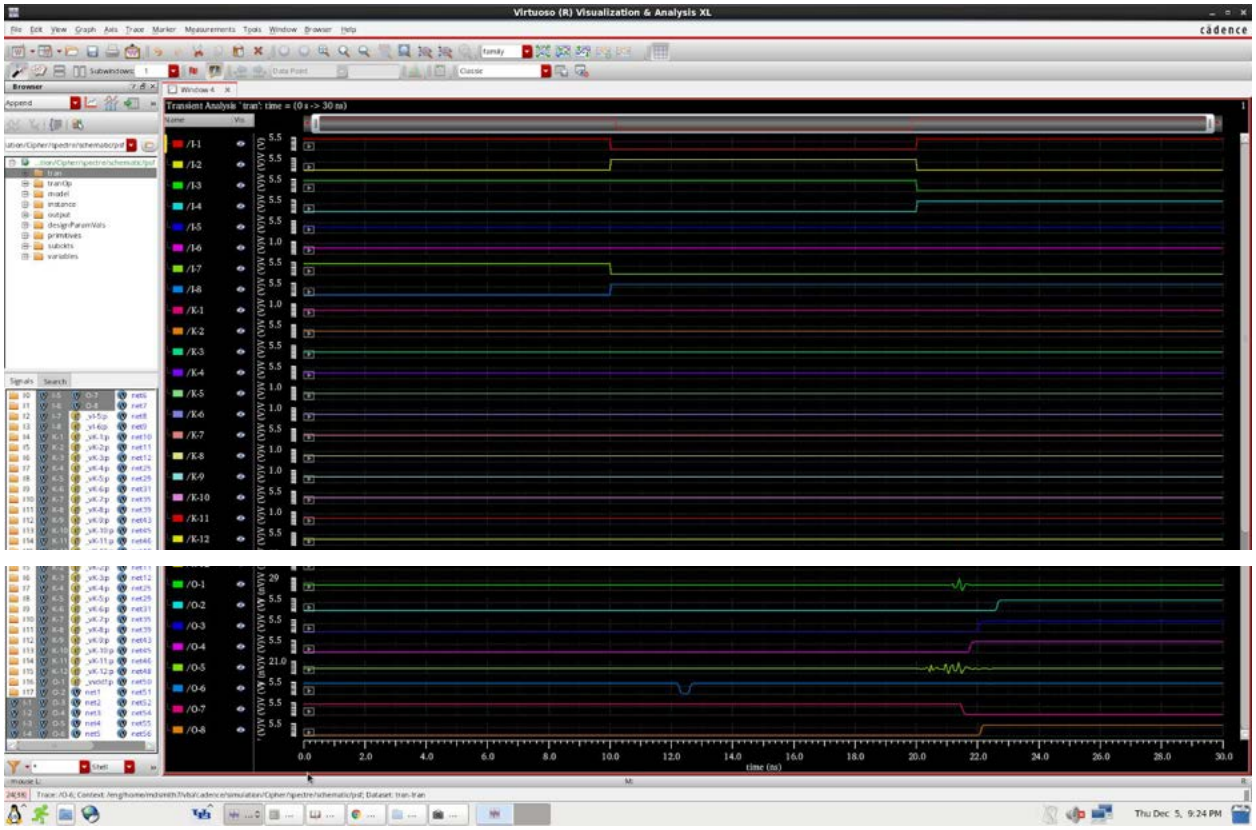


Figure 89: Schematic Simulation of the Cipher Function

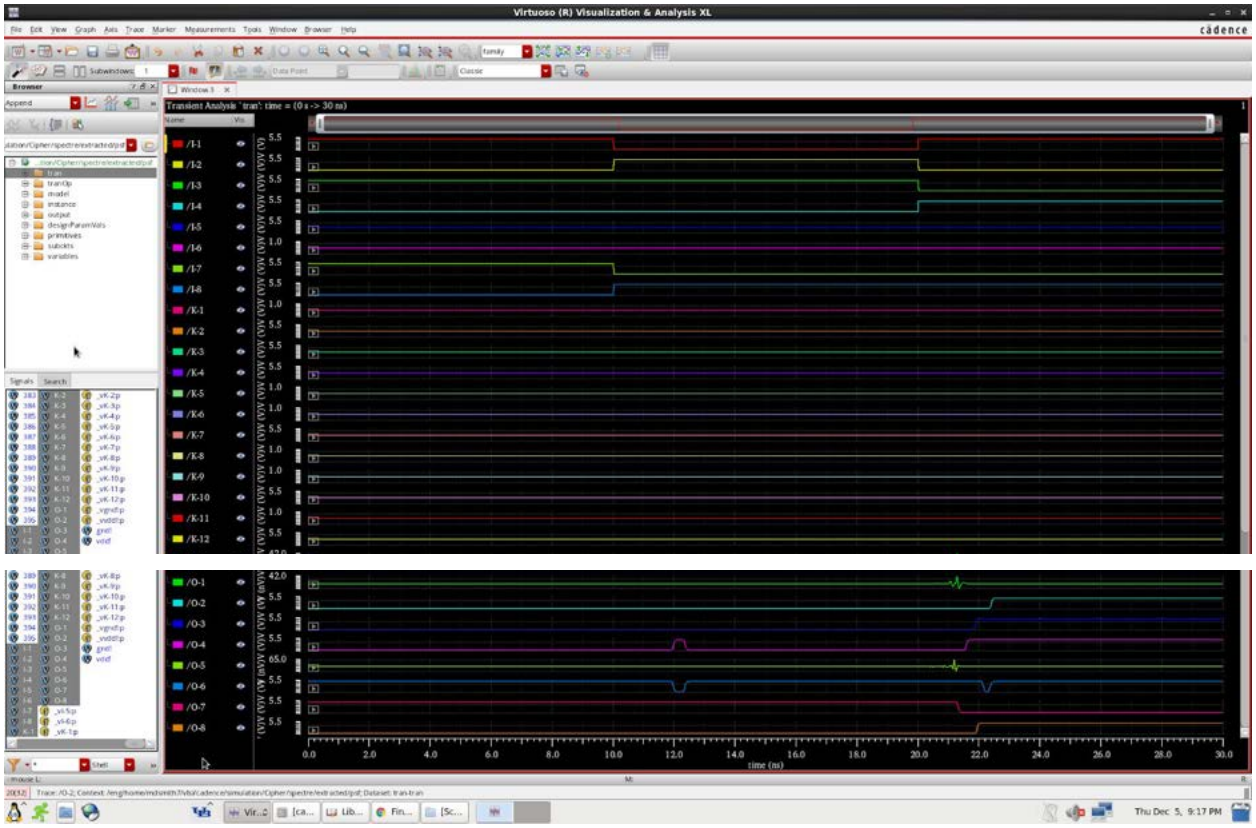


Figure 90: Extracted Simulation of the Cipher Function

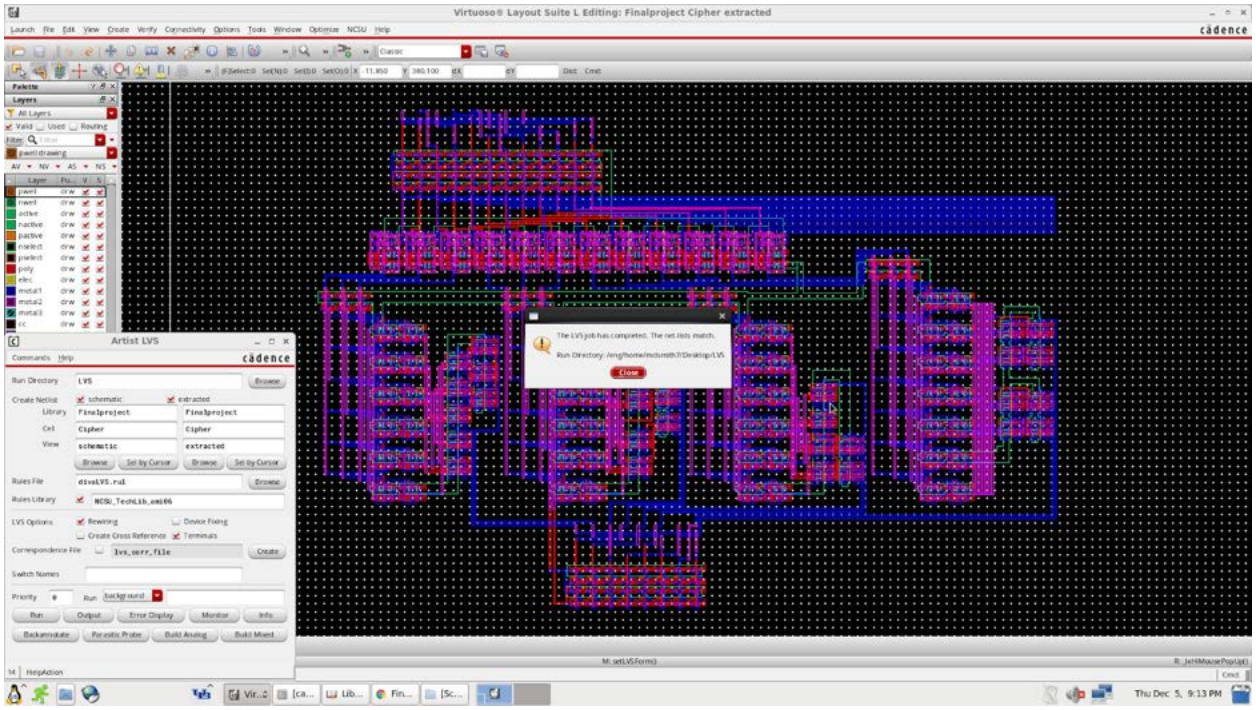


Figure 91: LVS Check for the Cipher Function

Round Generation - Views:

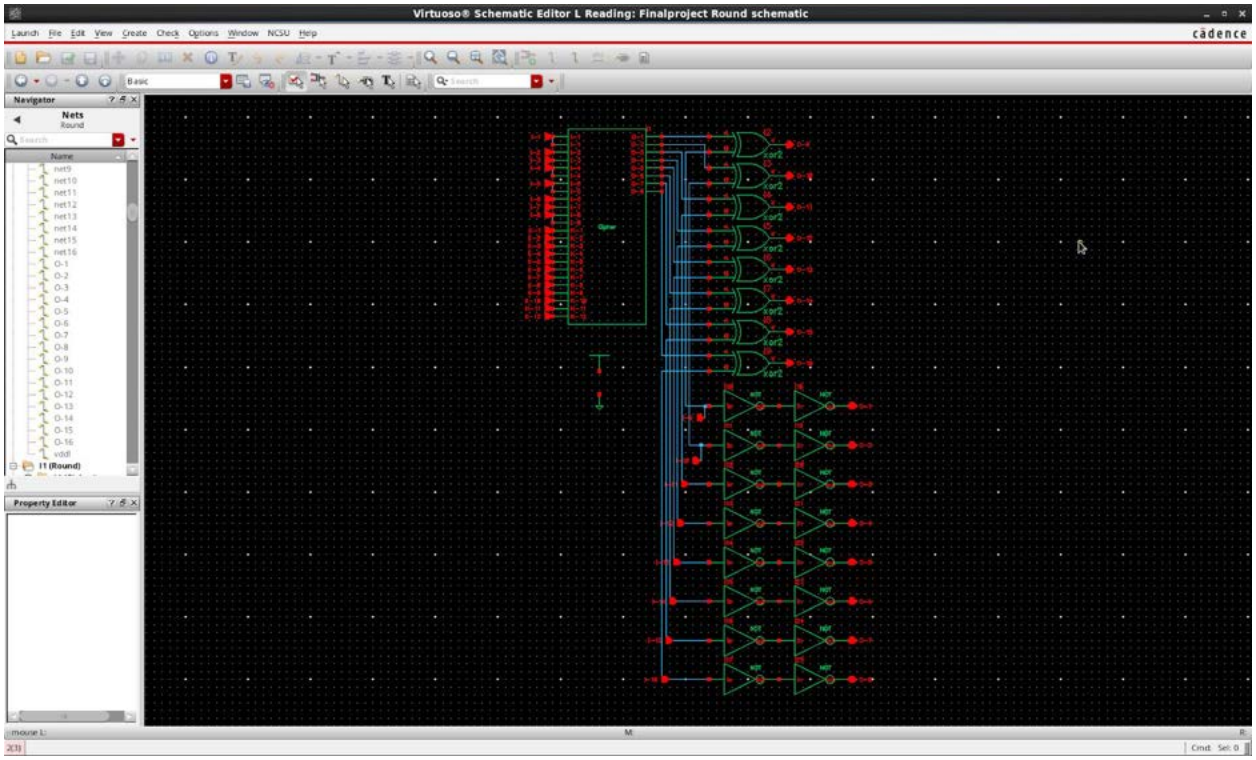


Figure 92: Schematic View of the Round Generation

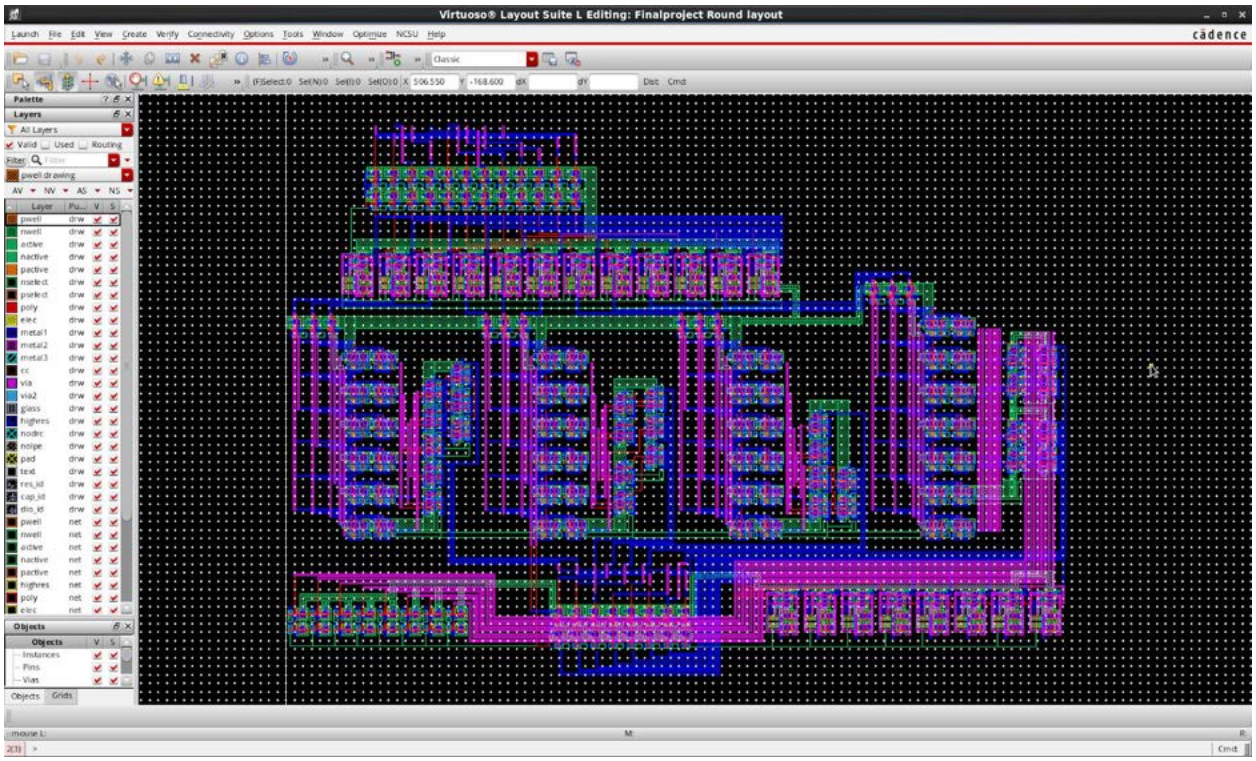


Figure 93: Layout View of the Round Generation

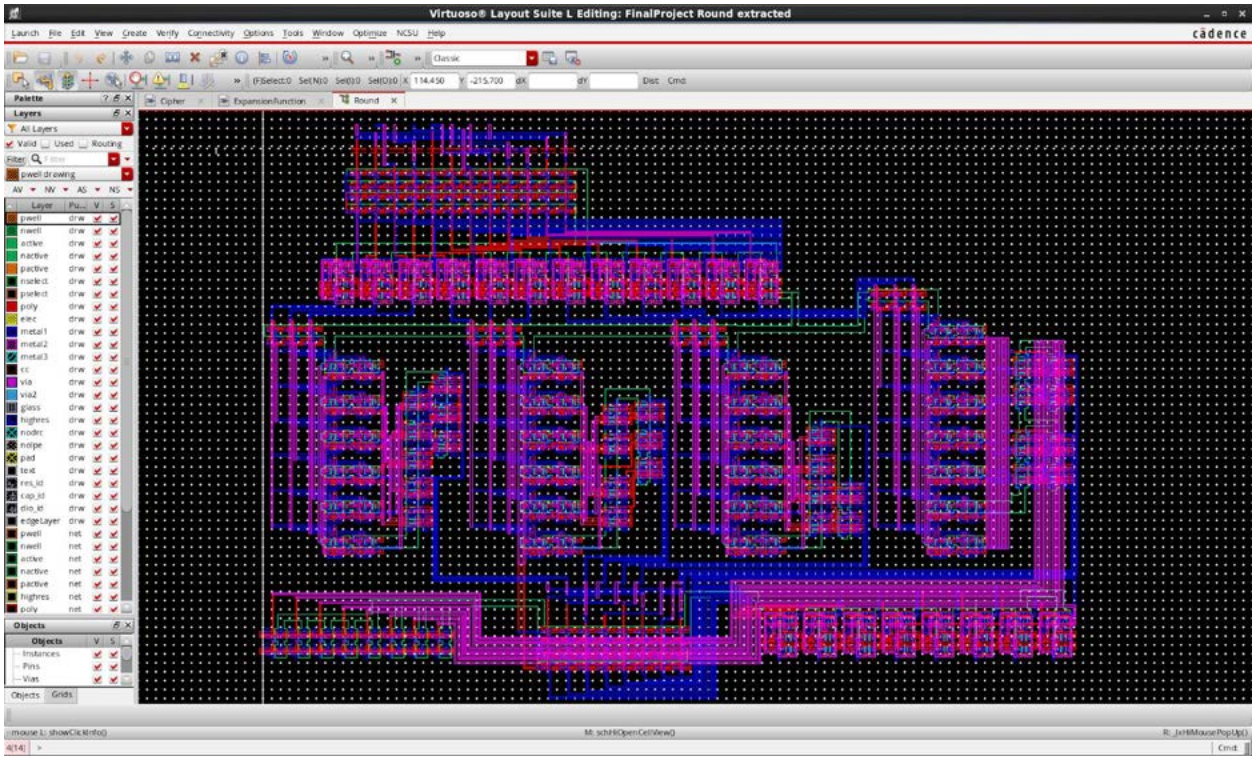


Figure 94: Extracted View of the Round Generation

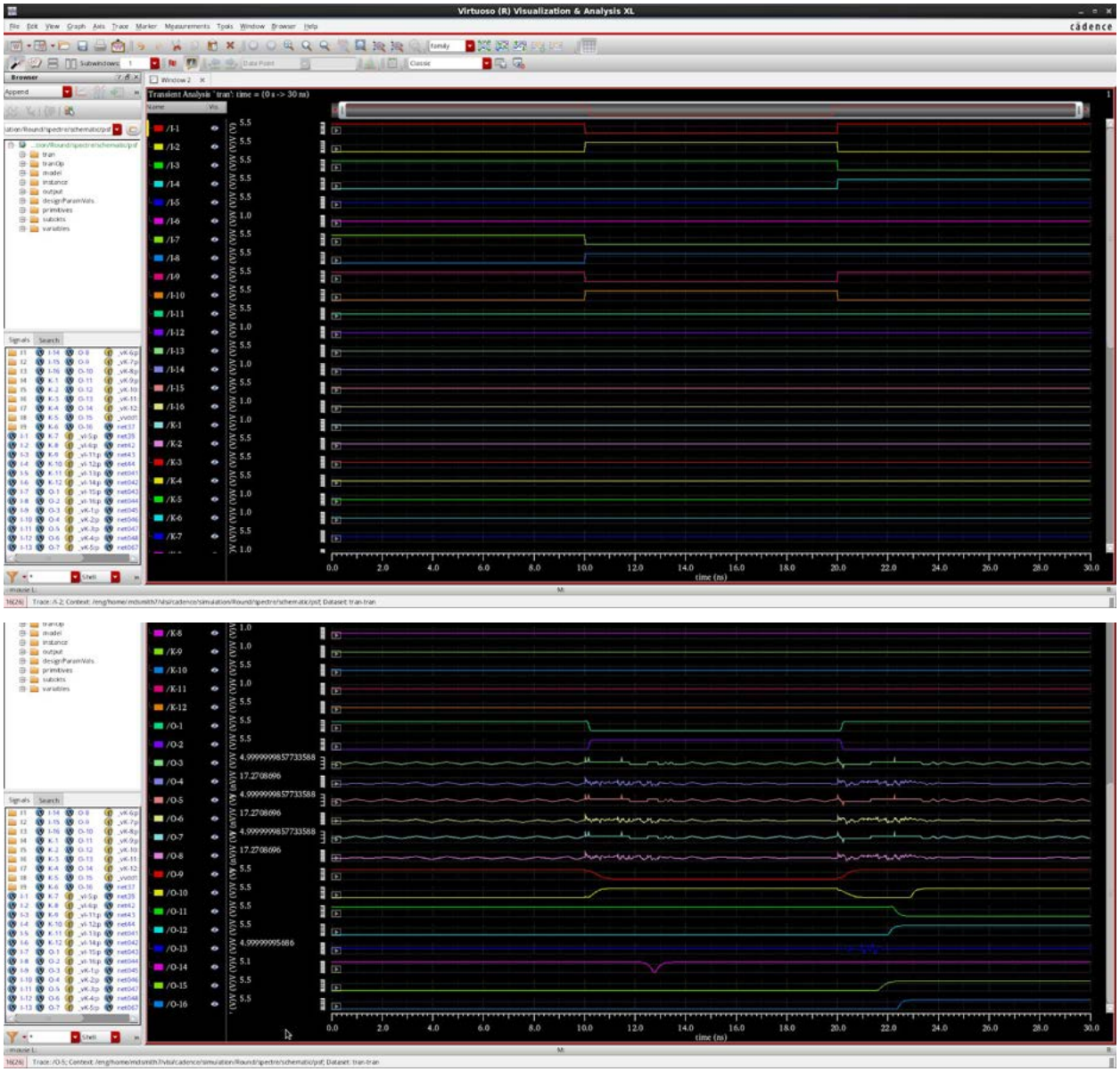


Figure 95: Schematic Simulation of the Round Generation

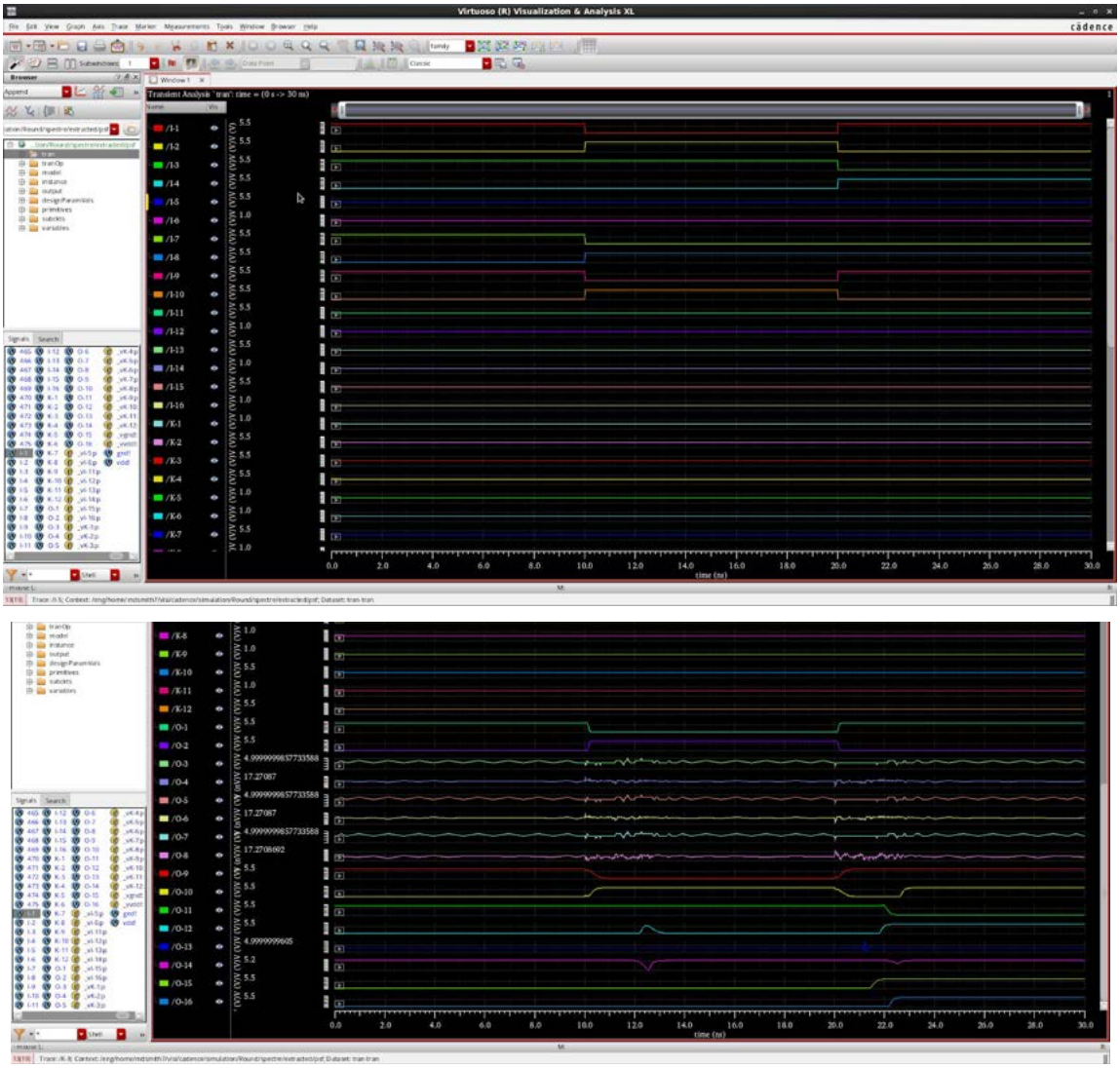


Figure 96: Extracted Simulation of the Round Generation

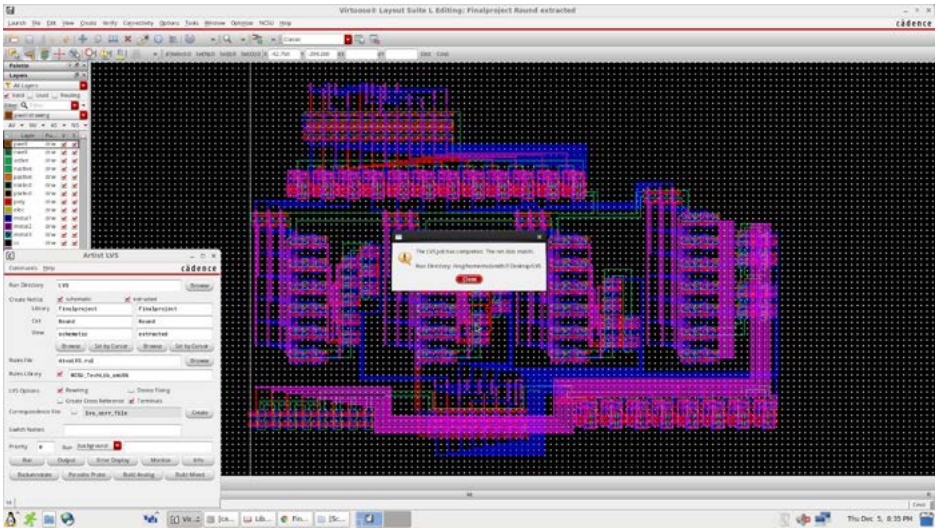


Figure 97: LVS Check for the Round Generation

Assembled Encryption - Views:

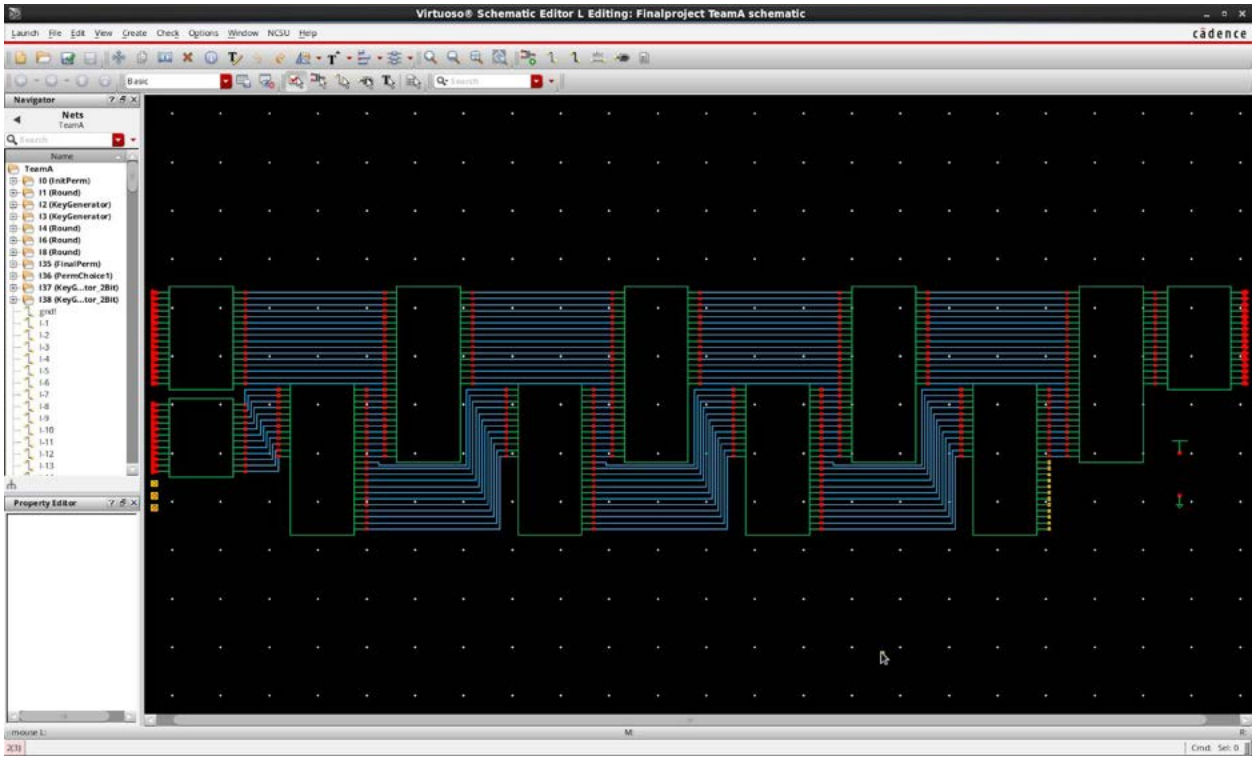


Figure 98: Schematic View of the Assembled Encryption

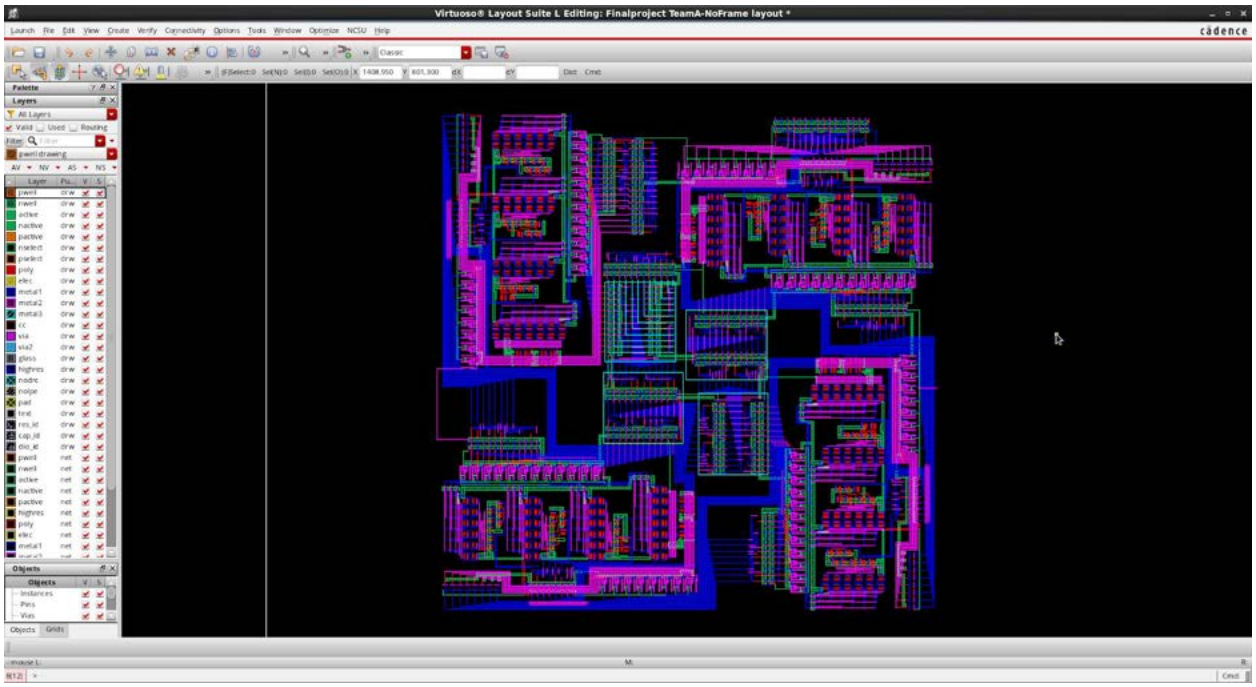


Figure 99: Layout View of the Assembled Encryption

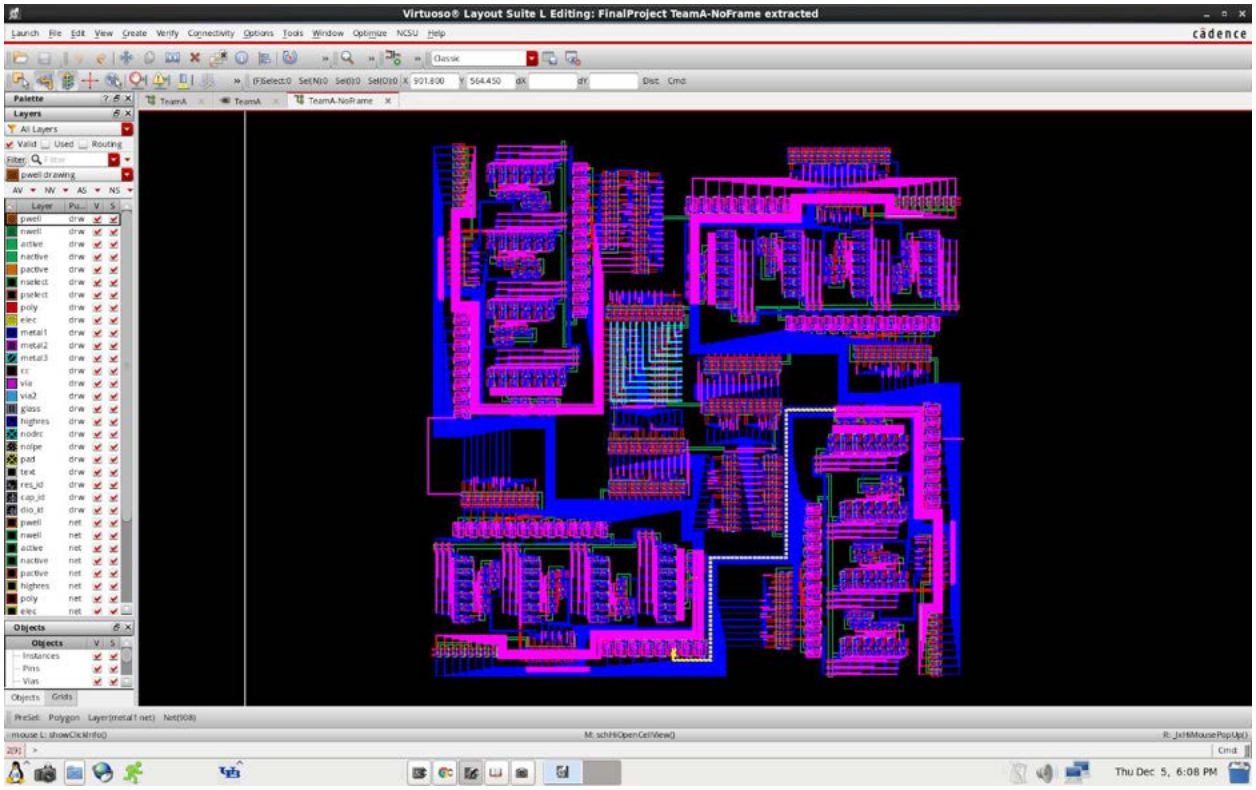


Figure 100: Extracted View of the Assembled Encryption

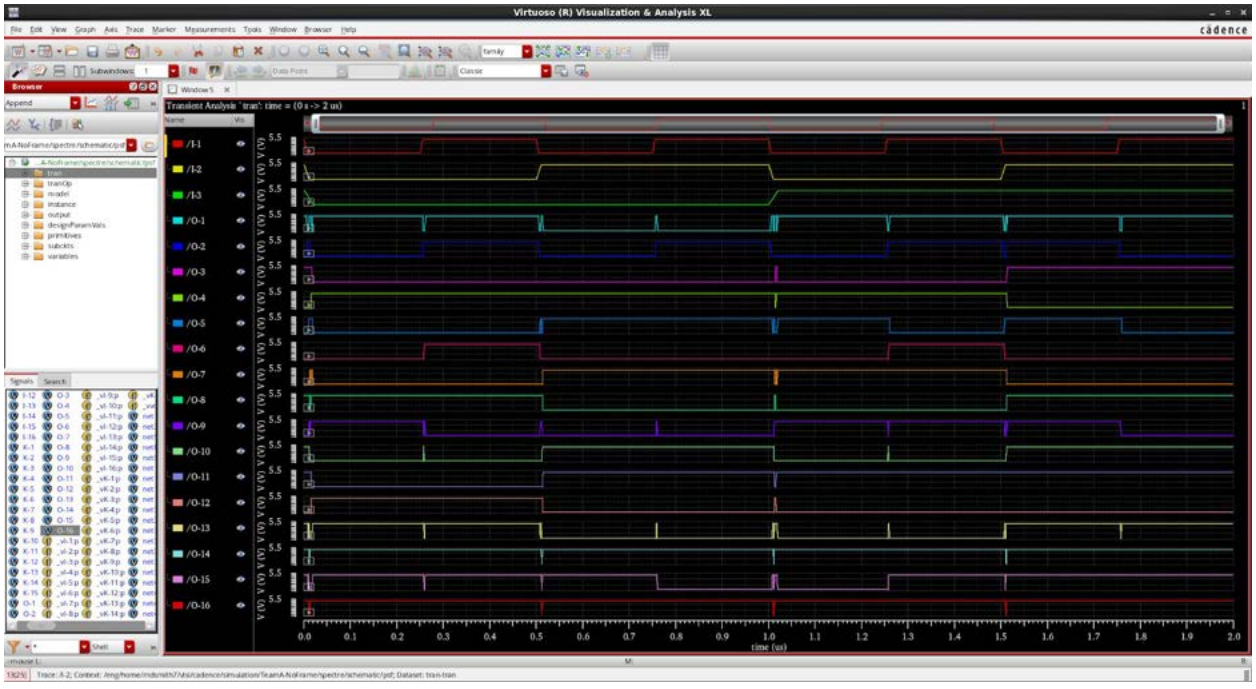


Figure 101: Schematic Simulation of the Assembled Encryption

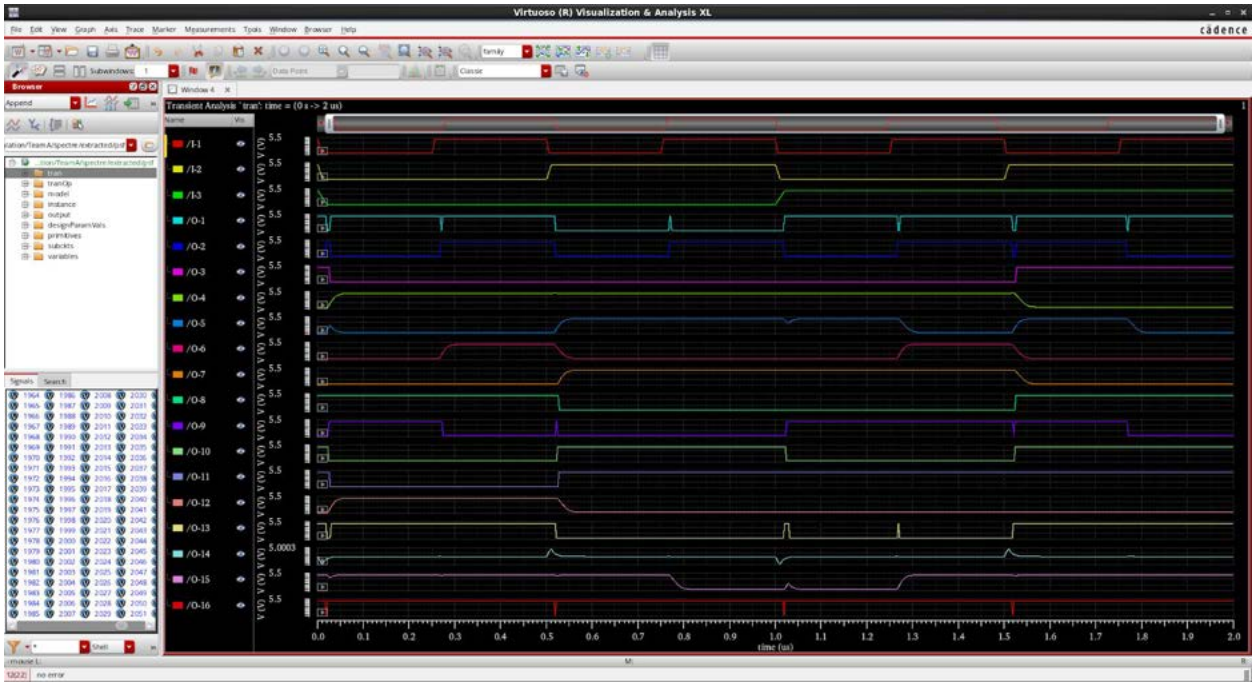


Figure 102: Extracted Simulation of the Assembled Encryption

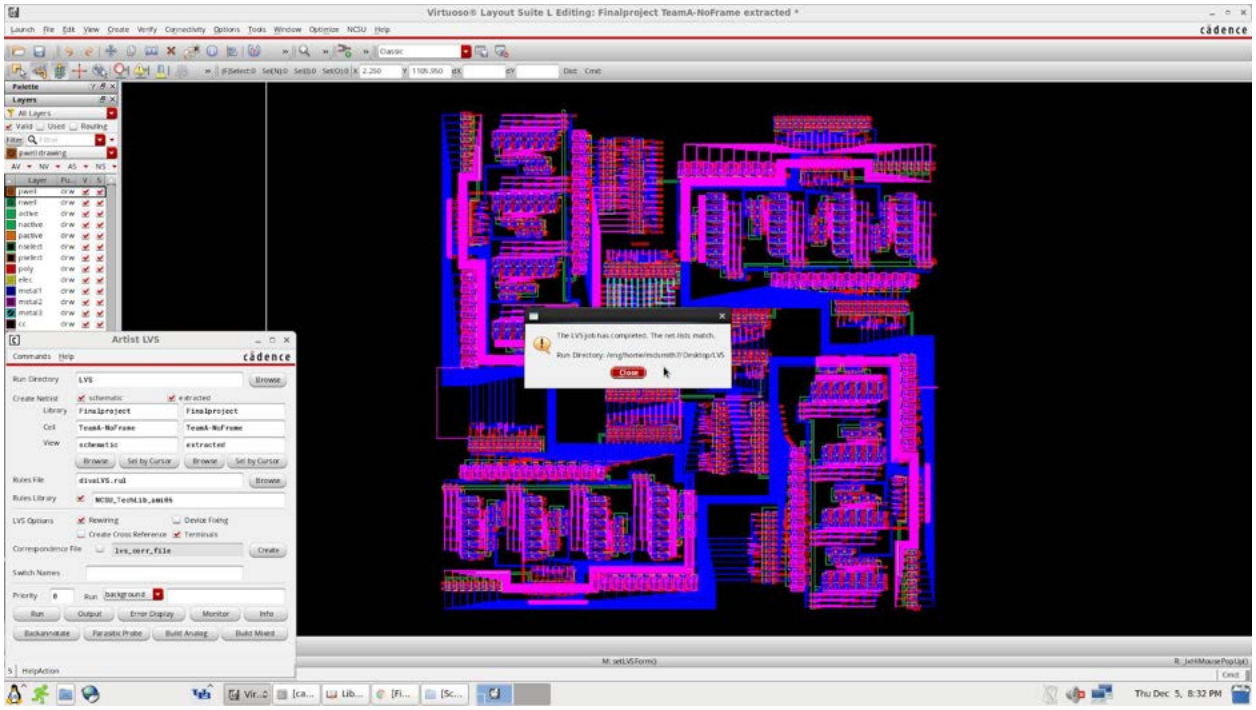


Figure 103: LVS Check for the Assembled Encryption

Assembled Encryption with Pad Frame - Views:

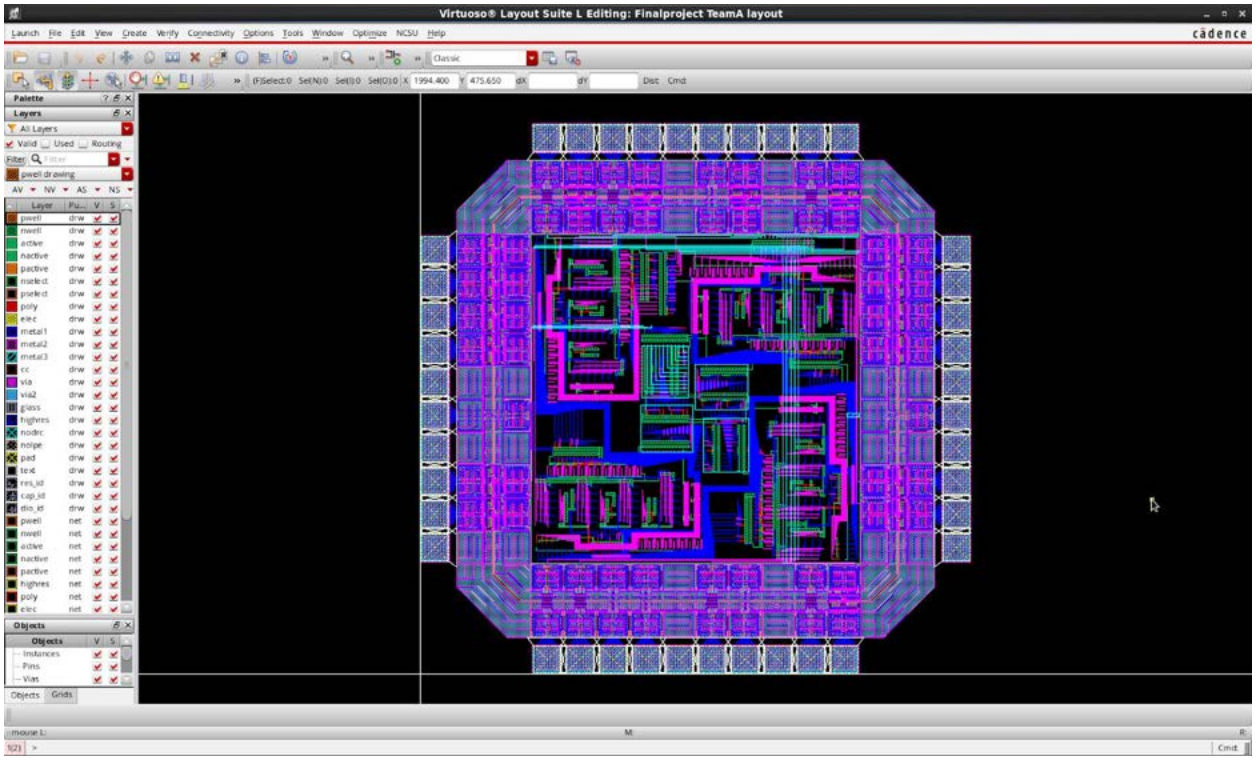


Figure 104: Layout View of the Assembled Encryption with Pad Frame

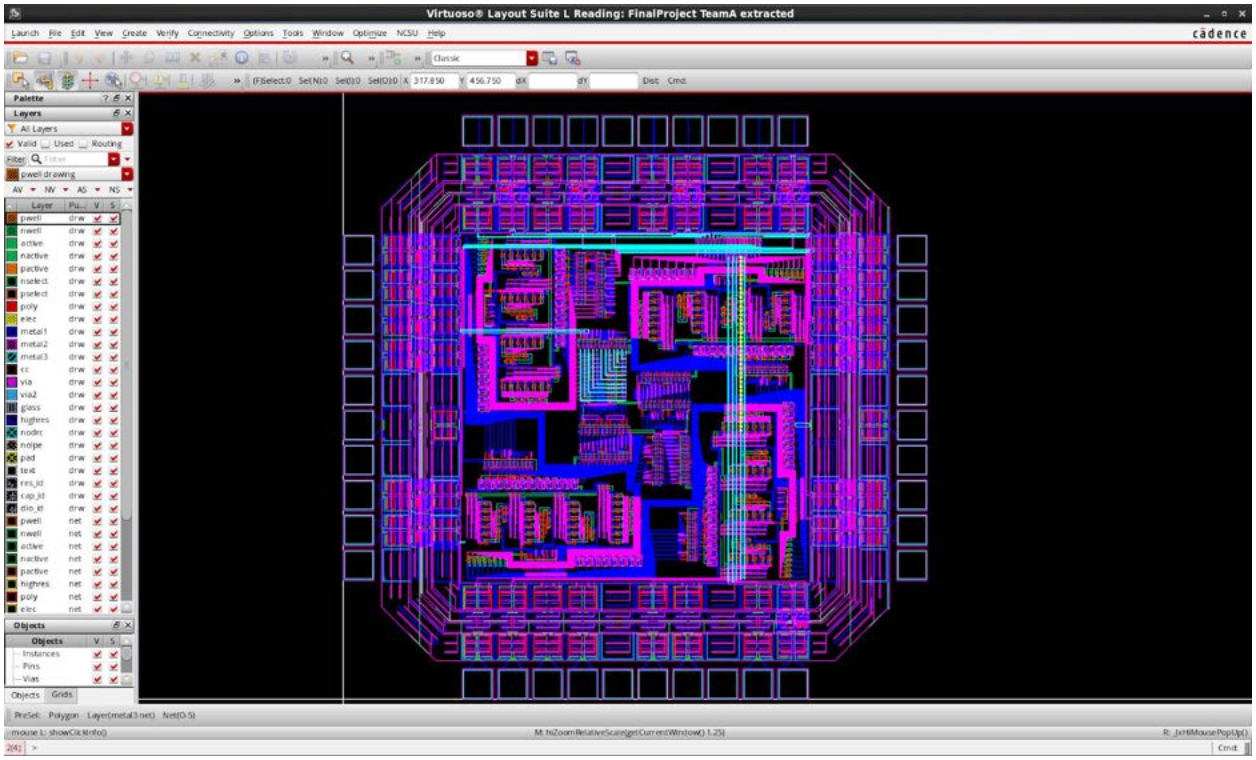


Figure 105: Extracted View of the Assembled Encryption with Pad Frame

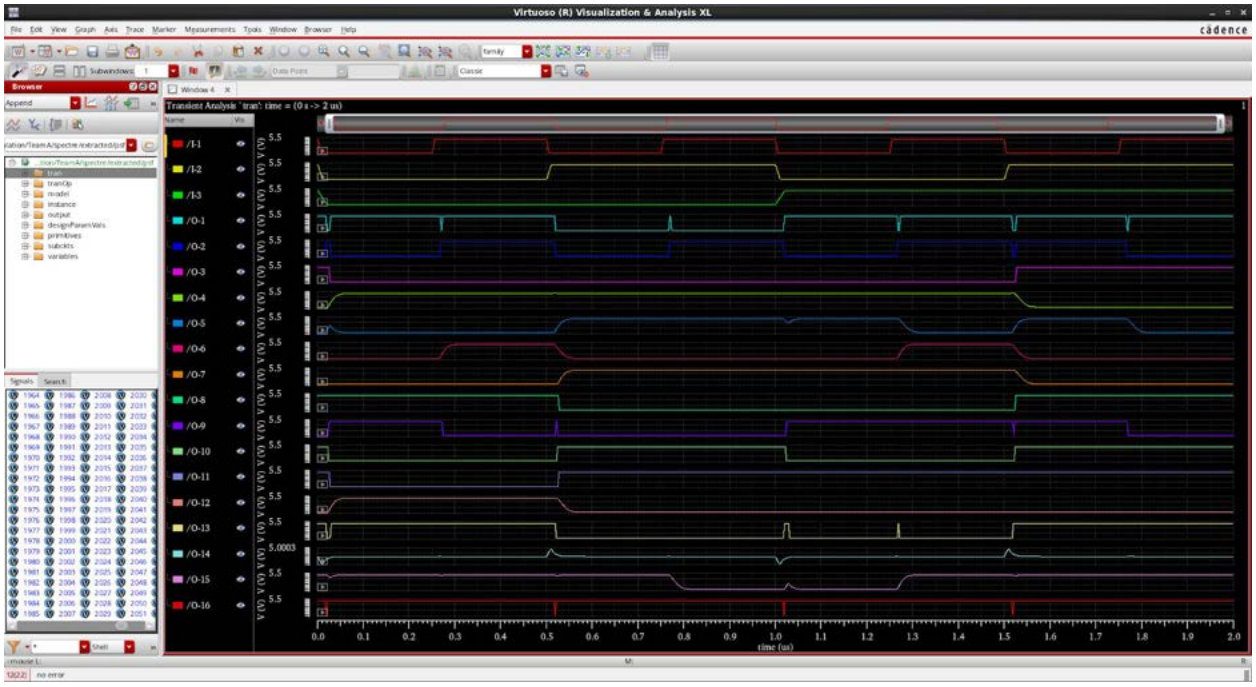


Figure 106: Extracted Simulation of the Assembled Encryption with Pad Frame

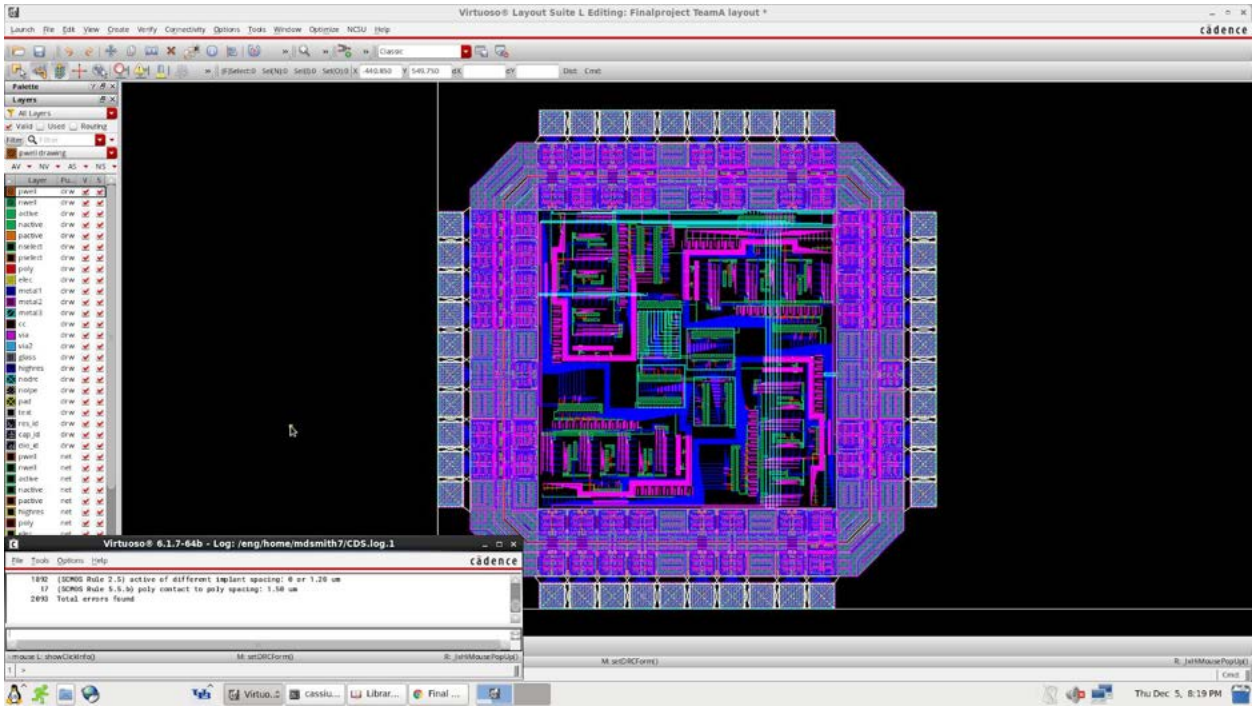


Figure 107: DRC Check for the Assembled Encryption with Pad Frame with 2093 errors

Flow Chart:

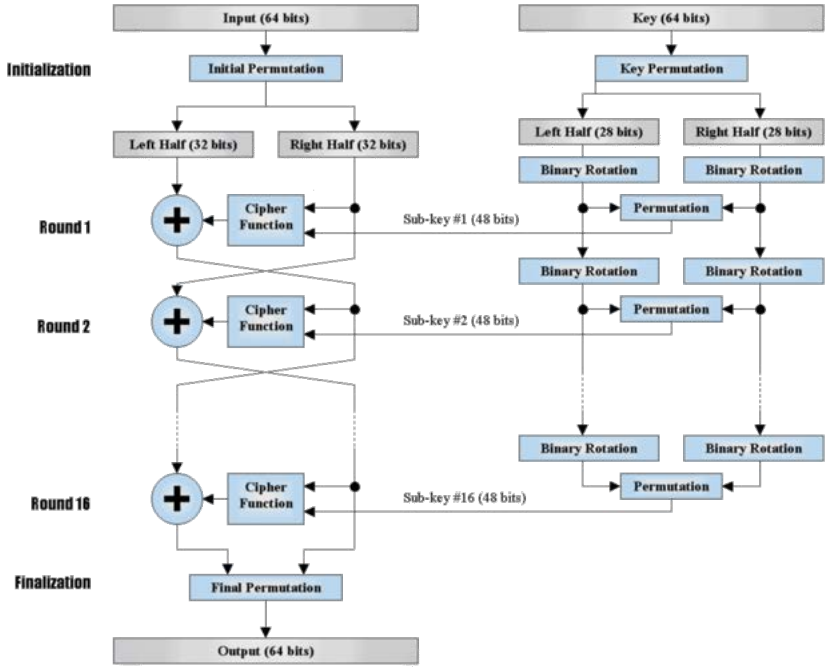


Figure 1: Process of Encryption Generation and Subkey Generation [1]

Work Distribution

Members	Work Distribution
Matthew Smith	Initial Permutation, Final Permutation, S-box 1, S-box 2, S-box 3, Assembly of the Encryption, Round Generation
Erik Vickerd	Circular Shift Left-1bit, Circular Shift Left-2bit, Expansion Function, Permutation, S-box 4, Cipher Function
Henry Dacres	Permutation Choice 1, Permutation Choice 2, Assembly of the Key Generation

Figure 108: Work Distribution among Group Members

Impact on Society:

Encryption Algorithms are used in all facets of modern society, Many understand the importance of keeping items secure and protected but many fail to understand just how it is used in daily life. Cryptography has a big impact on the global economy, and your daily life. These algorithms protect your most sensitive data whether it be messages, bank account information, documents or a plethora of other sensitive information. Aside from daily life, in industry encryption is used to protect sensitive information, like customer information, trade secrets, and financial records. Critical infrastructure, like power plants, use encryption to protect their systems from malicious attackers. DES for a long time served as an industry standard. Although it has been obsolete for many years it was used to encrypt top secret government documents. All these years later DES still serves historically as one of the most influential encryption algorithms.

Conclusion:

The DES algorithm played a pivotal role in cryptography throughout the 1970s, 80s and 90s, where it was the most widely used encryption algorithm. DES encryption was the standard to compare the security of other algorithms to. Although important in its day, advancements in computing allowed the relatively short 56 bit key to be easily cracked, so the algorithm became insecure as a standard, and therefore obsolete. Due to this, it was replaced by AES encryption. Our 16-bit implementation of this algorithm is a good demonstration of how this encryption algorithm works in a scaled-down manner.

References:

All References in IEEE Format

[1] H. Megahed, "DES (Data Encryption Standard)," Cybrary.IT, 26 January 2016. [Online]. Available: https://www.cybrary.it/0p3n/des-data-encryption-standard/. [Accessed 2019].

[2] "Data Encryption Standard (DES)," in Cryptography and Network Security, Pearson, p. 32.

[3] "Wikipedia," Wikipedia, September 2019. [Online]. Available: https://en.wikipedia.org/wiki/DES_supplementary_material. [Accessed 2019].