Thao tác trong file excel

- Dùng công cụ chuyển đổi từ dạng “string” qua “Hexadecimal“

Đổi chuỗi sang mã hex

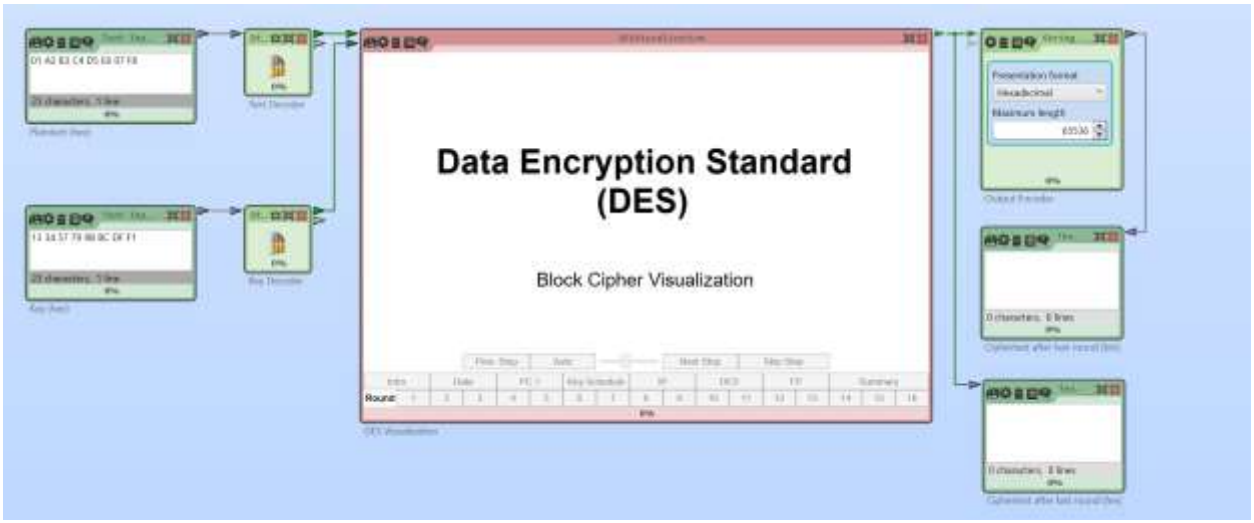
Chuỗi	Hello wo	Hex	48656C6C6F20776F

- Có được dữ liệu từ ”input” và “Key” => có được mã (đã được mã hoá)

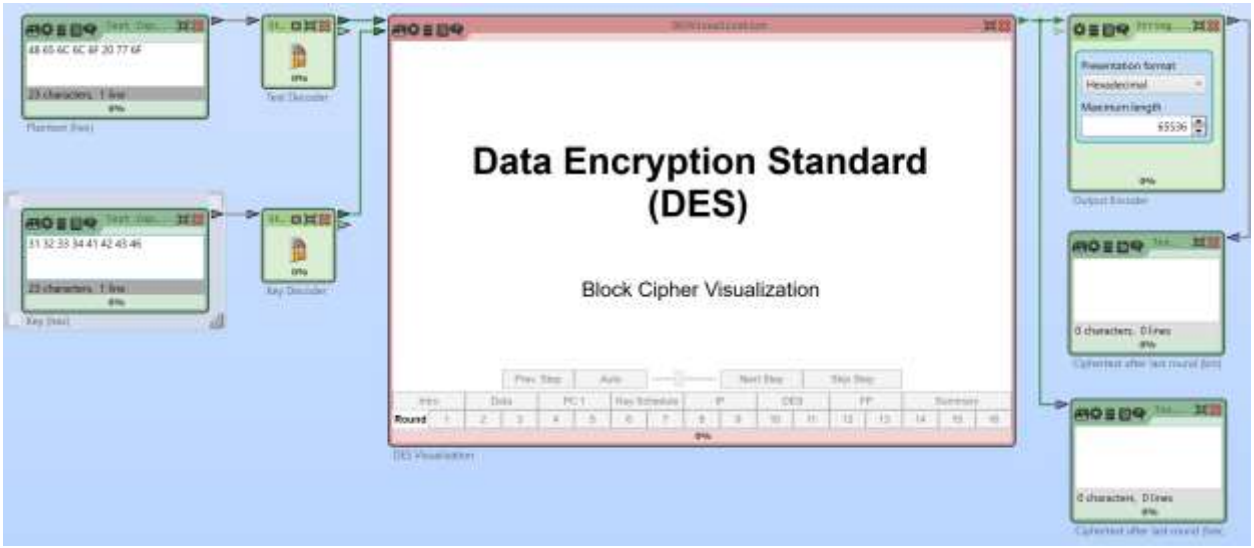
INPUT (kí tự)	Hello wo
KEY(Mã HEX)	3132333441424346
MÃ	0A 30 66 5C 8A 5D 01 1F

Thao tác trong Cryptool

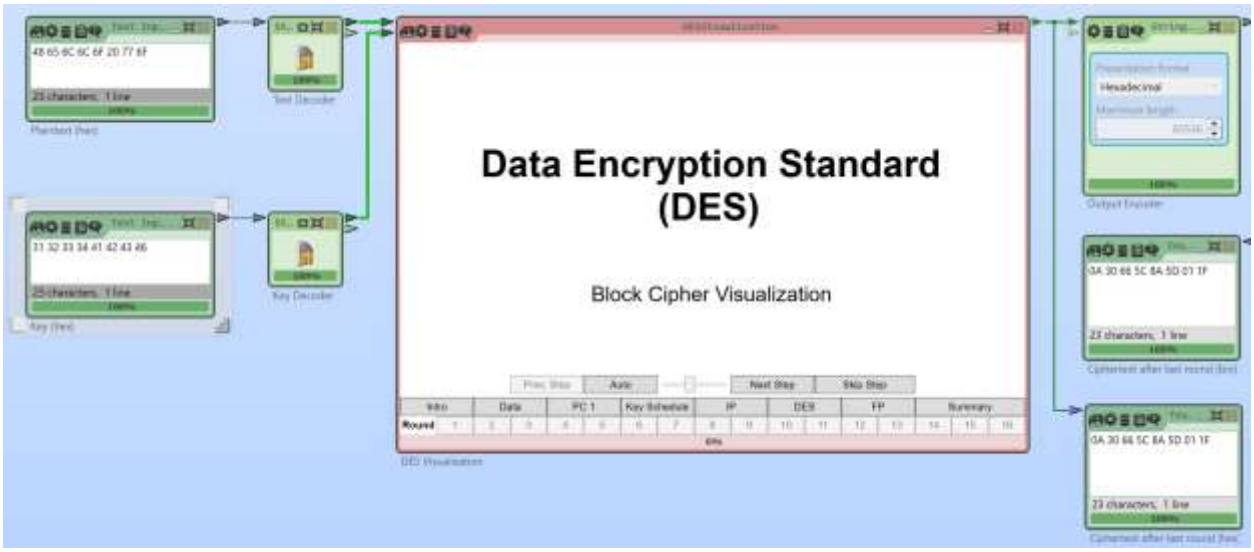
- Giao diện:



B1: lấy input đầu vào (ở dạng Hex) đưa vào plaintext và lấy khoá chính (Key)

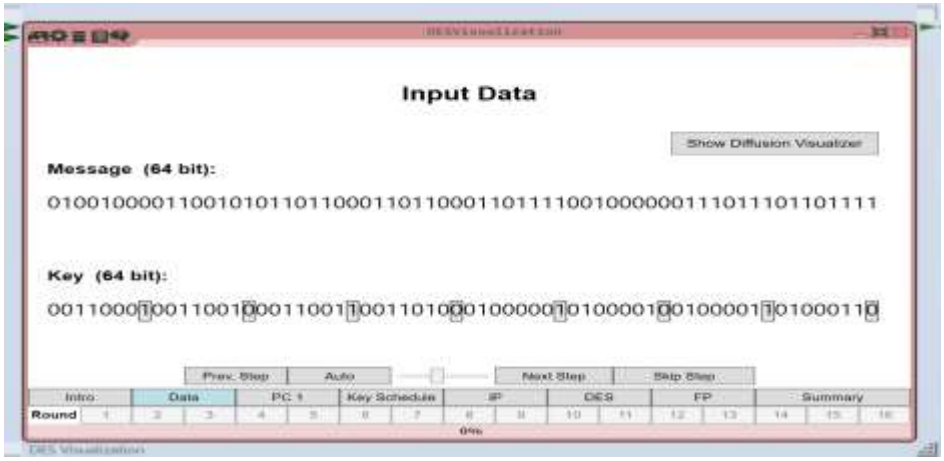


B2: Nhấn play chạy chương trình, nó sẽ trả ra mã đã được mã hoá

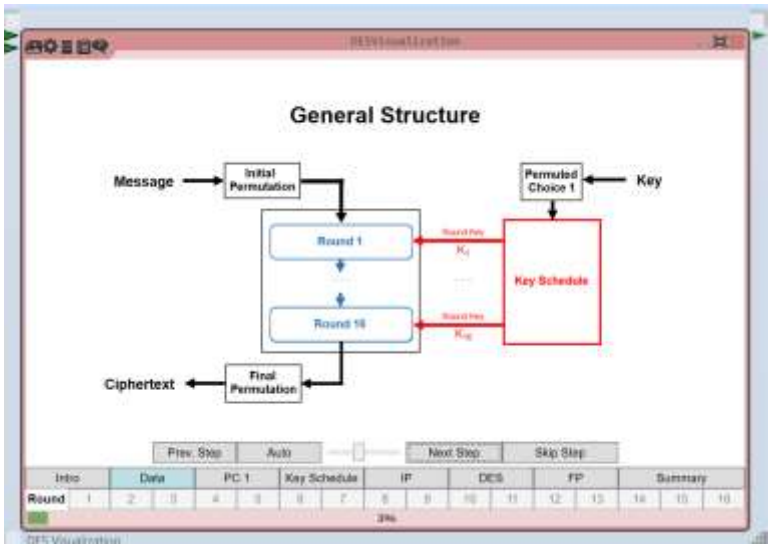


- Quan sát rõ hơn các bước
- Đối với Key

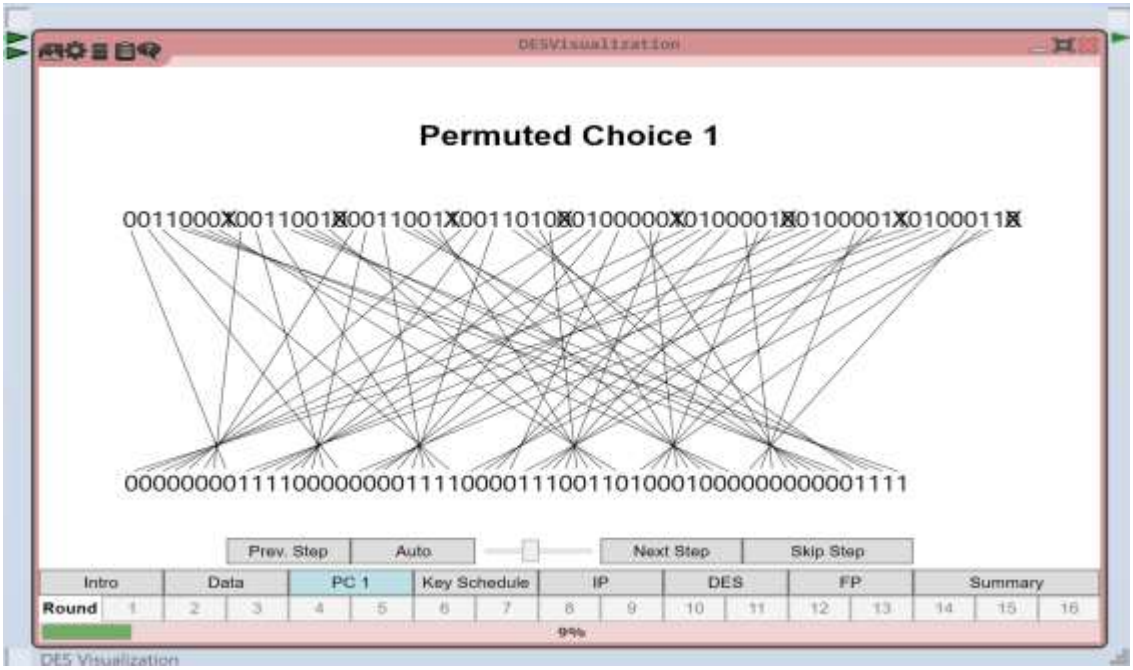
+ dữ liệu đầu vào khi được chuyển từ Hexa sang binary



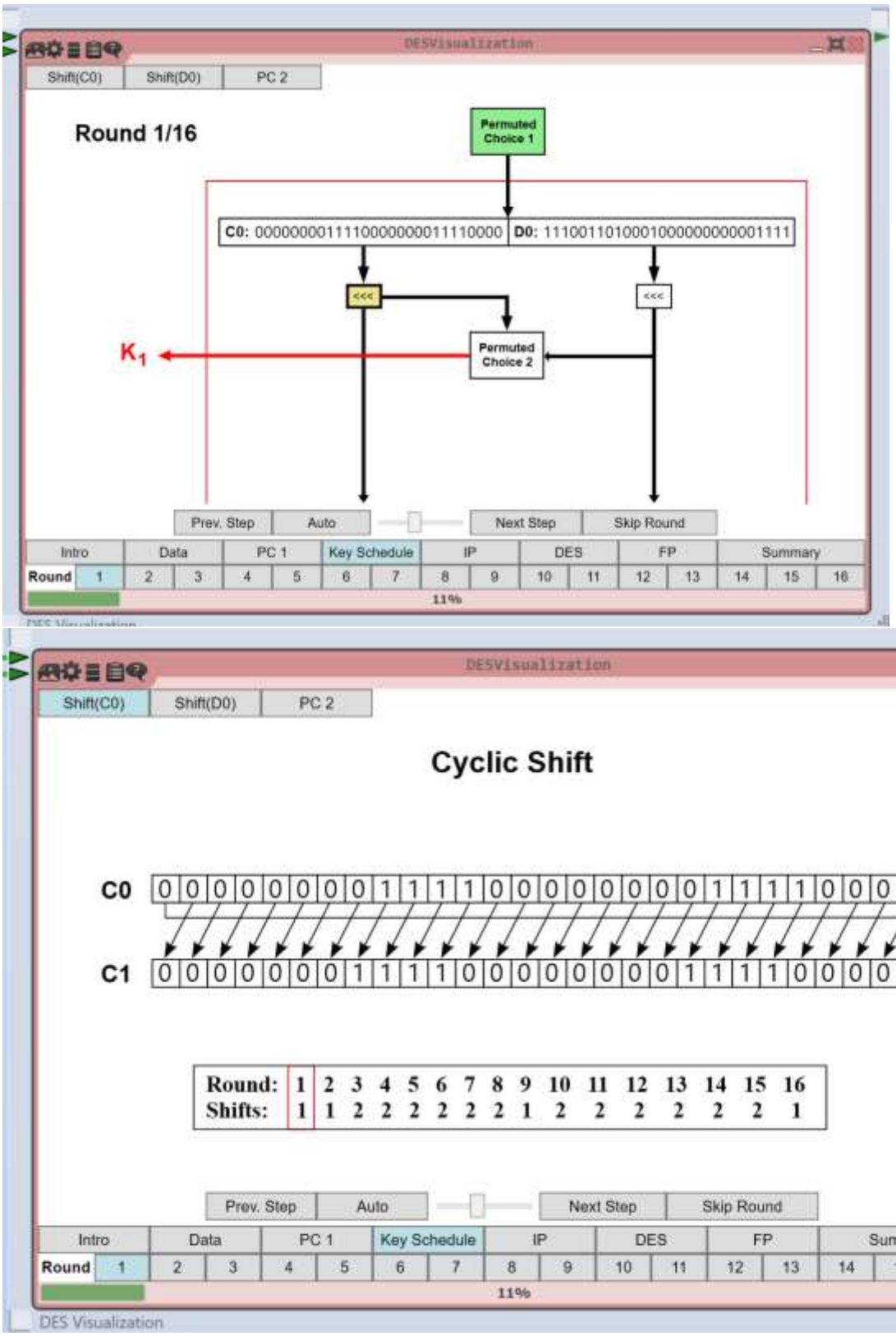
- Sơ đồ



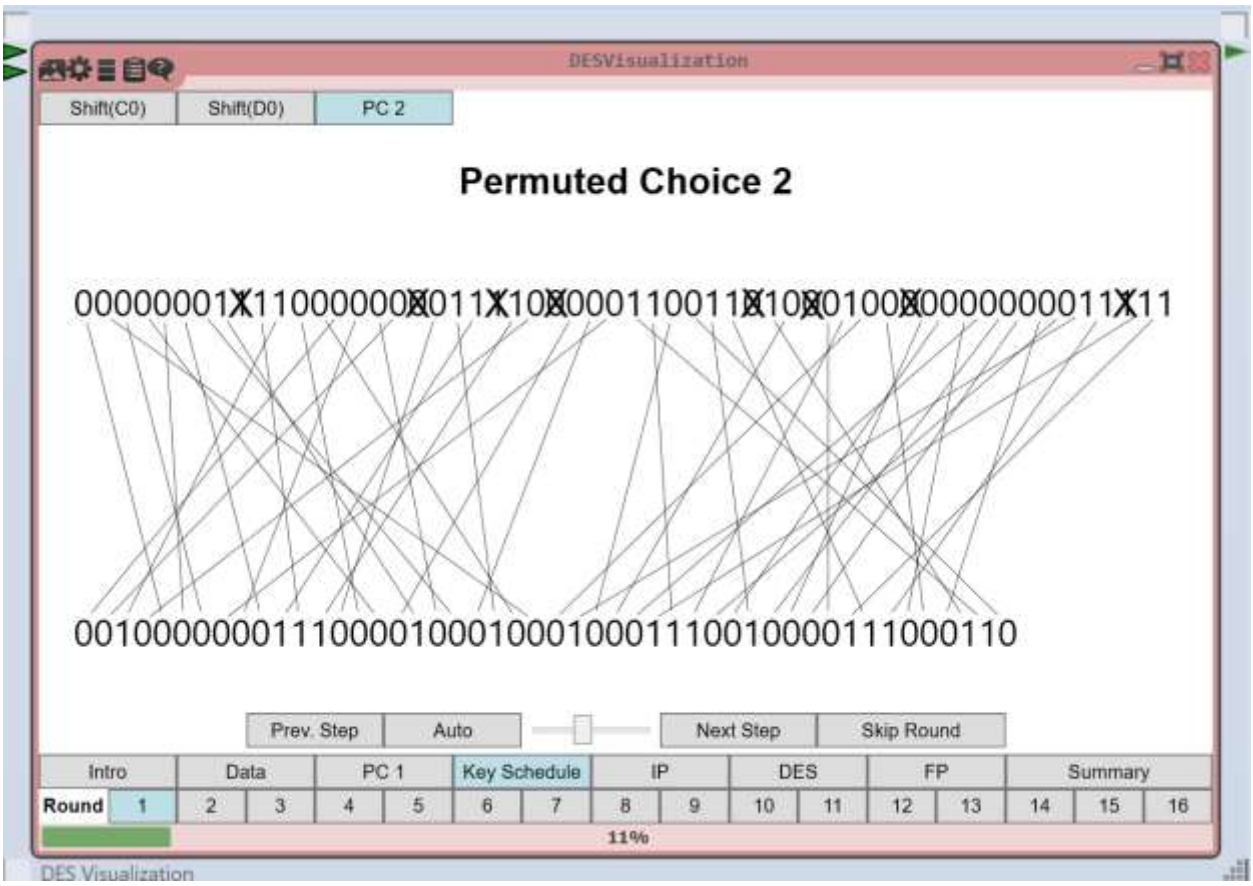
- Ánh xạ qua bảng PC1 sẽ thu được PC1



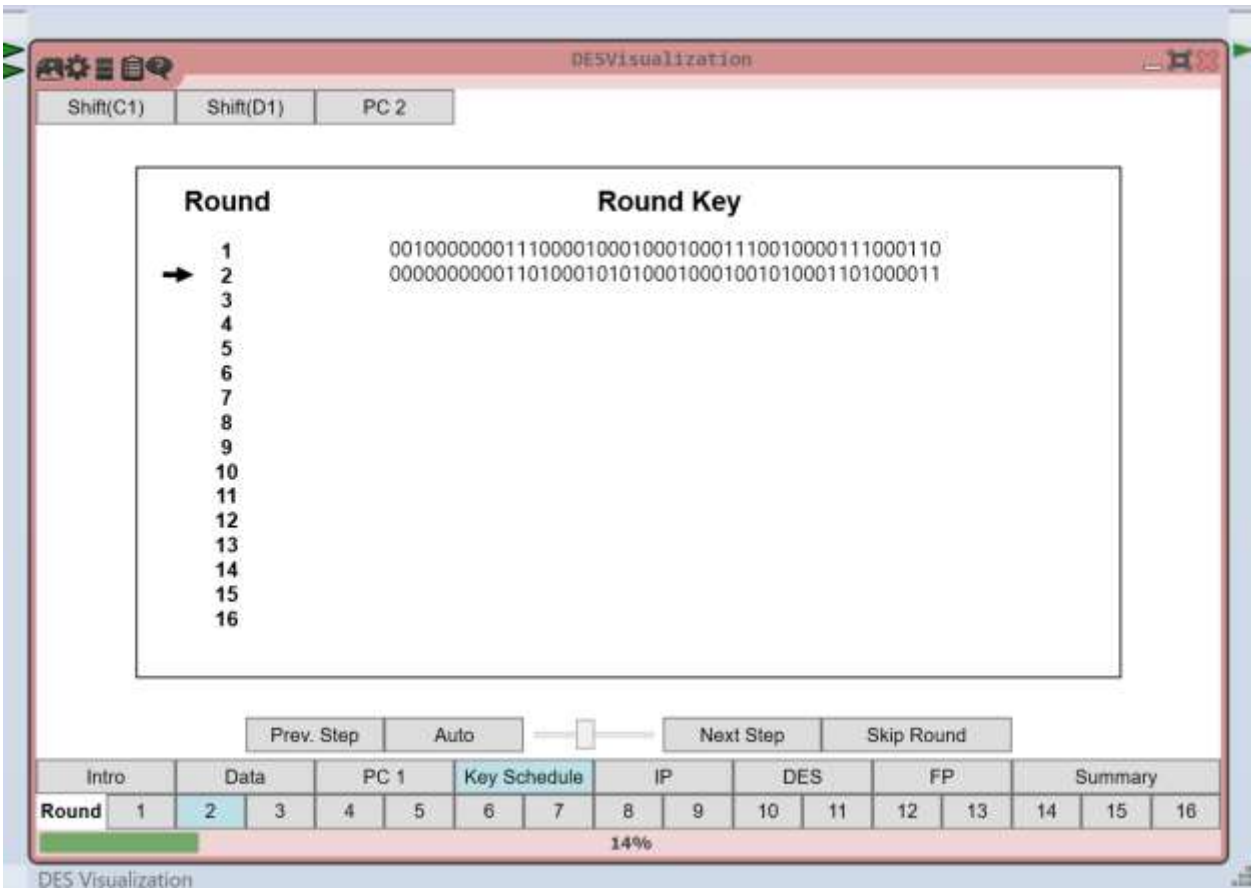
- Có PC1, dữ liệu được chia thành 2 nửa và bắt đầu dịch 1 bit cho vòng khoá 1 và vòng khoá 2 (ở cả 2 nửa D và C)



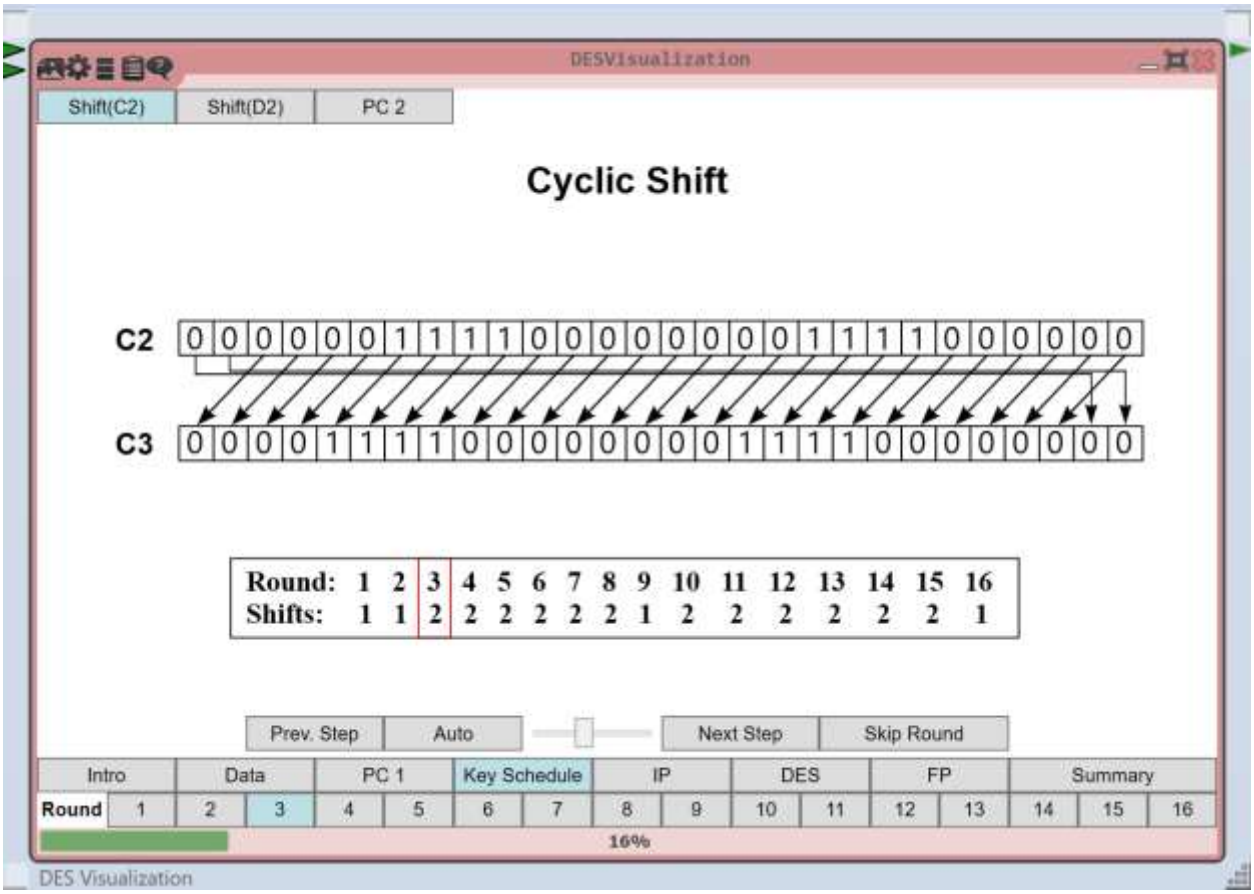
- Sau khi dịch 1 bit thì dữ liệu đi qua hoán vị PC2 sẽ thu được PC2



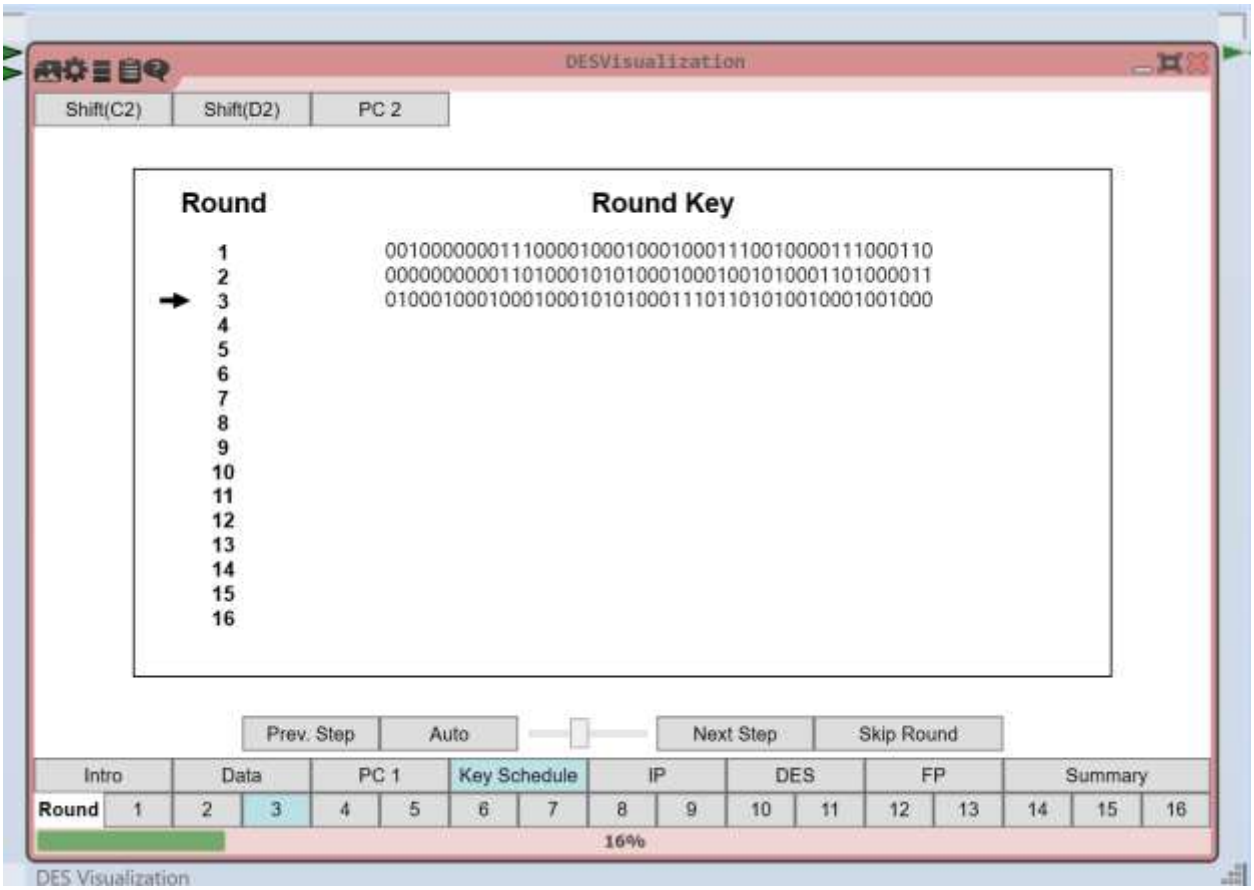
Sau đó tạo ra khoá con K1 và tiếp bước thứ 2 sẽ tạo ra khoá con k2



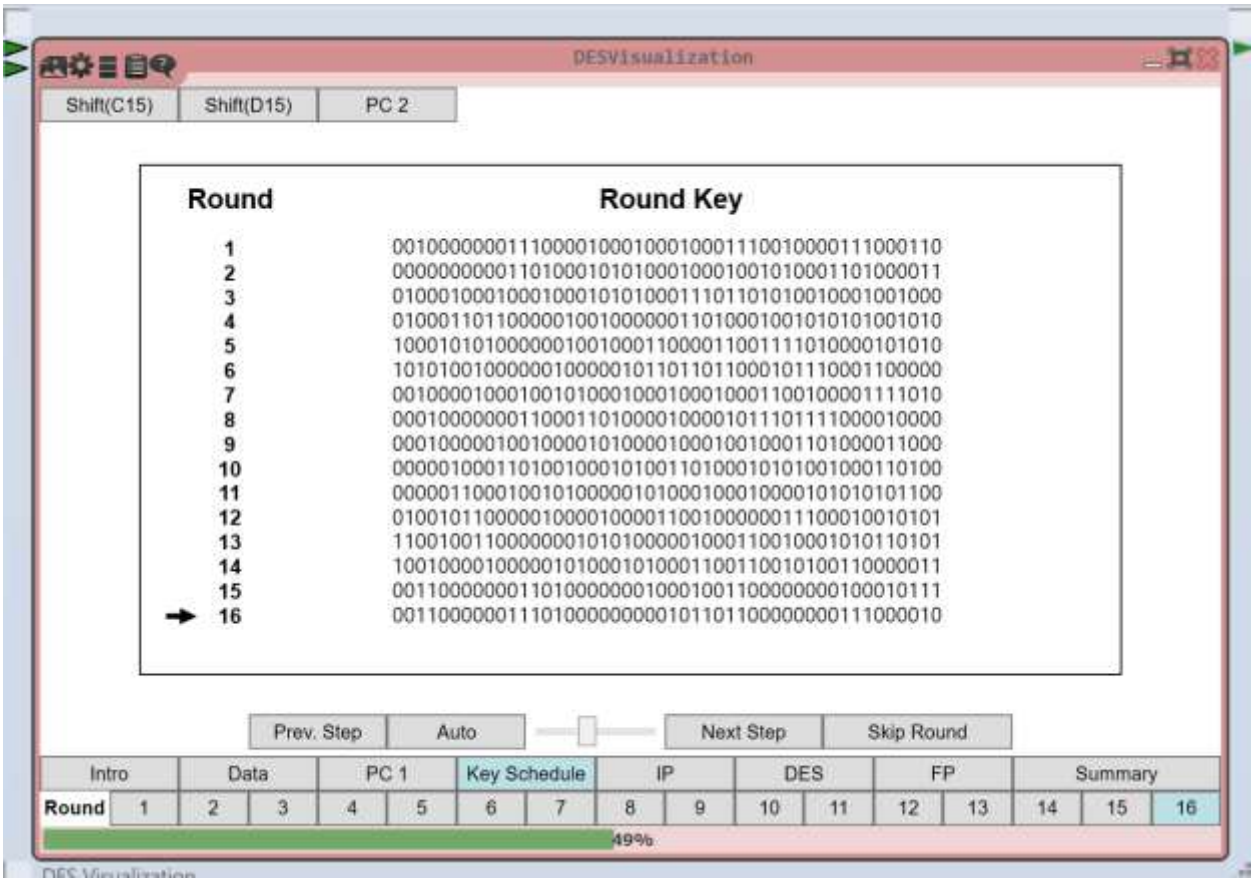
- Ở vòng thứ 3 tương tự với các bước trên nhưng khi dịch bit thì chương trình sẽ dịch 2 bit (cho 2 nửa D và C)



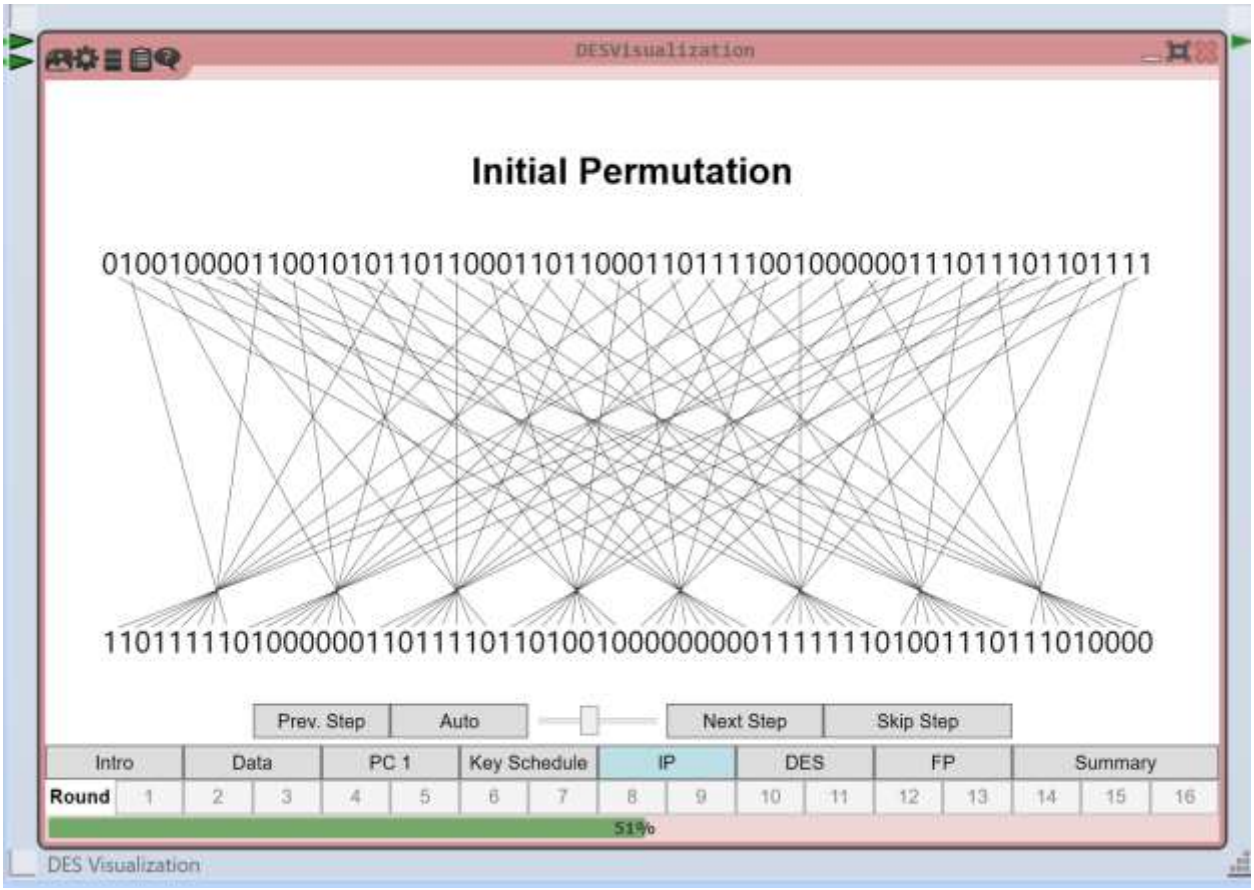
- Sau đó 2 nửa được gộp lại để trải qua hoán vị PC2 và tạo ra khoá con K3



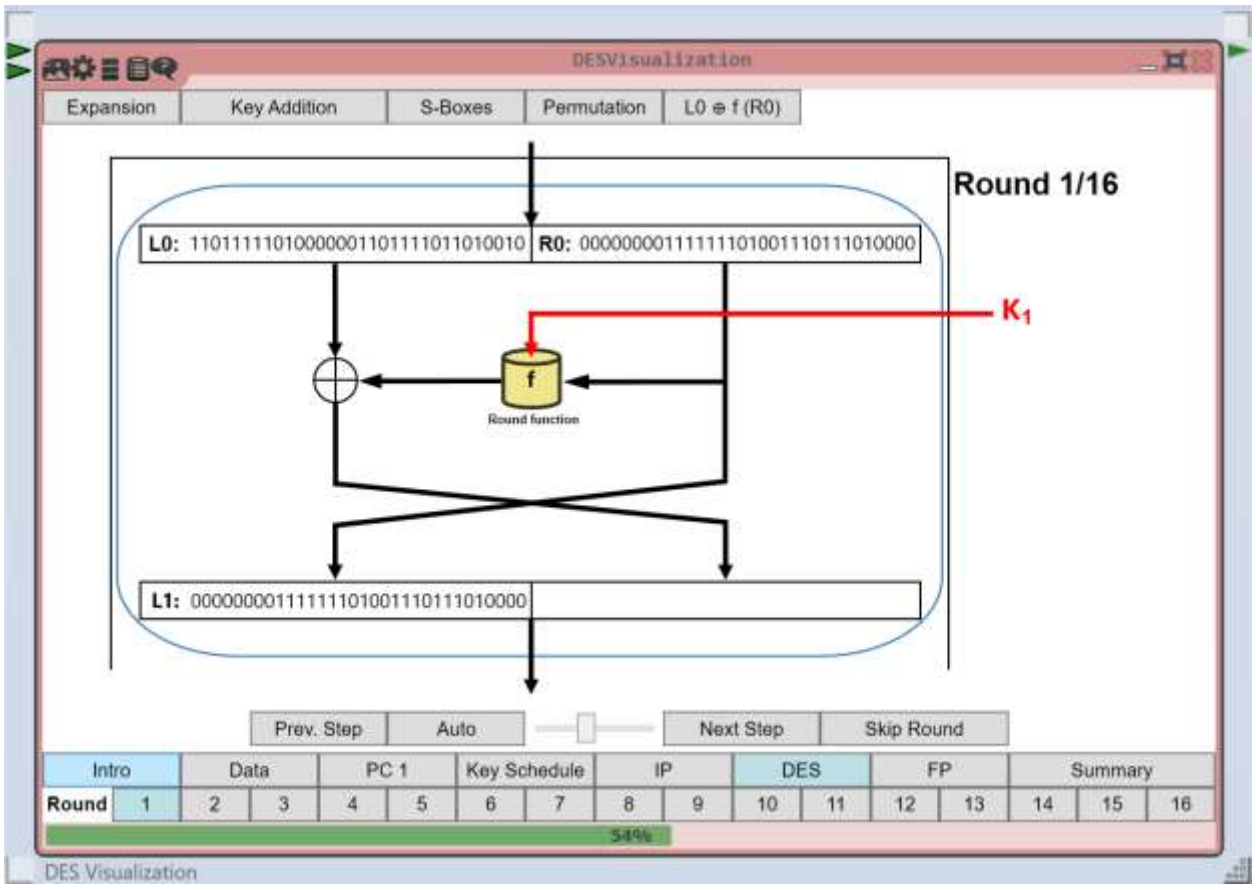
- Tiếp tục các bước trên cho tới khi có đủ 16 khoá con



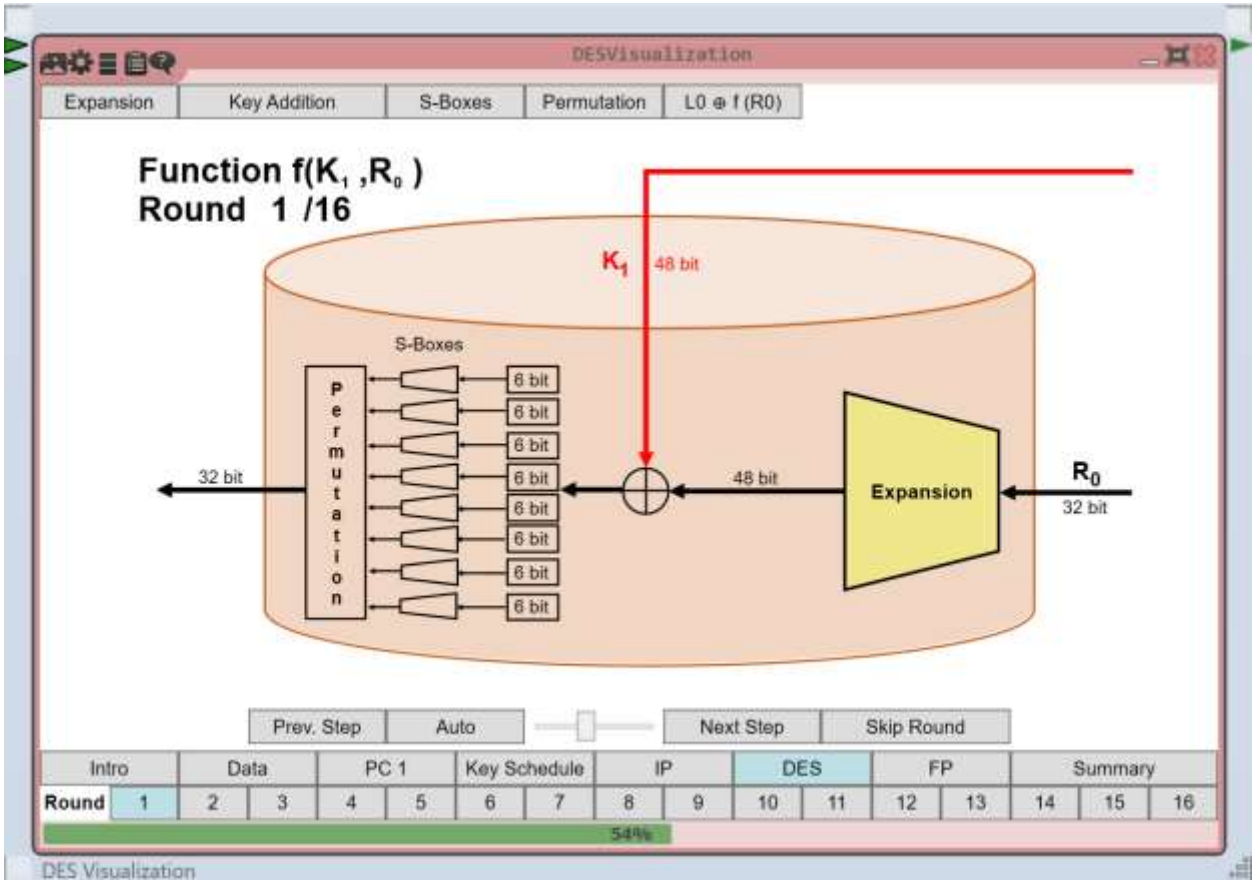
- Đối với khối đầu vào (Plain text)
  + Khối binary ban đầu sẽ đi qua hoán vị IP ,thu được IP



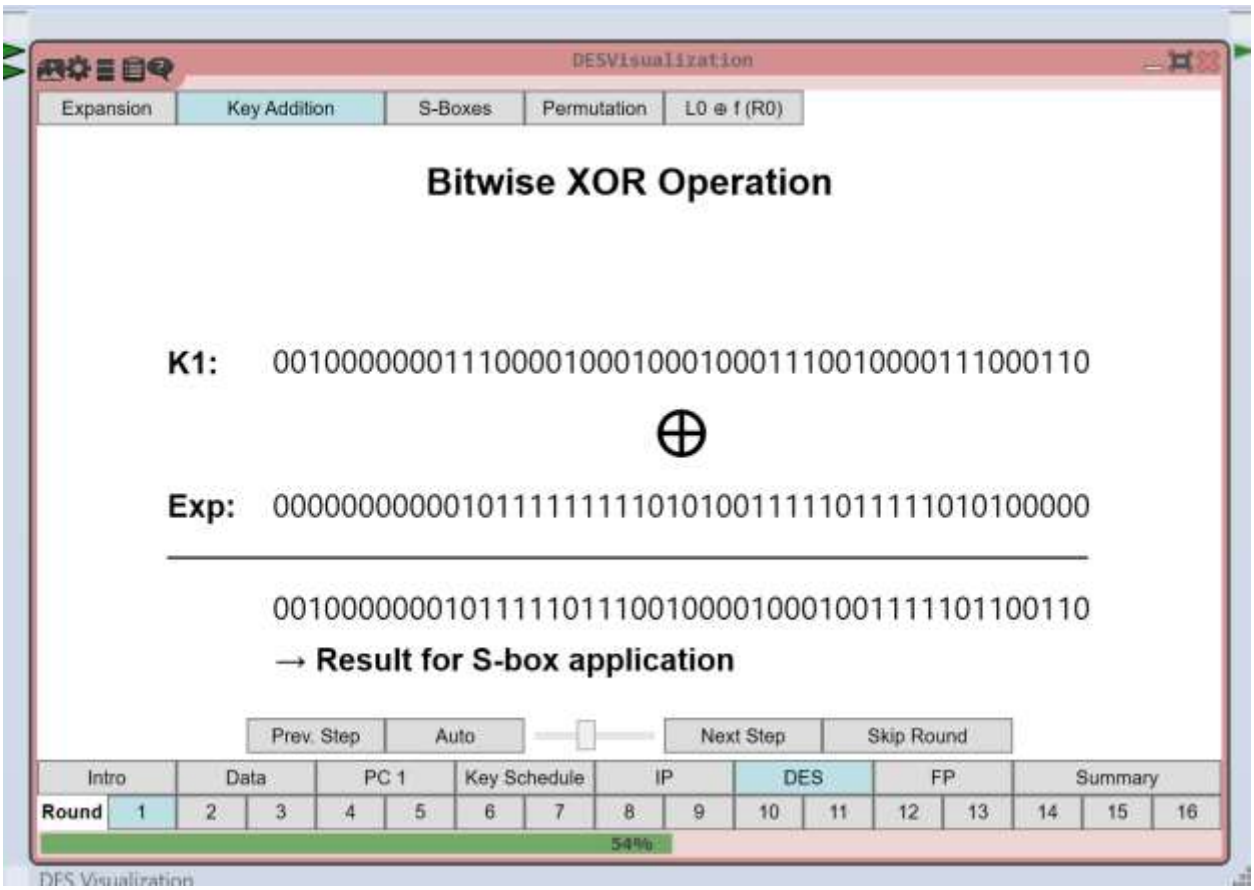
- Sau đó dữ liệu được chia thành 2 nửa (trái và phải)
  1. Nửa bên phải sẽ được chuyển thẳng xuống nửa trái làm giá trị mới và nửa bên phải cũng đi vào hàm Feistel (cục chữ f)


DESVisualization
Expansion
Key Addition
S-Boxes
Permutation
L0 ⊕ f (R0)
Round 1/16
L0: 11011111010000001101111011010010
R0: 00000000111111101001110111010000
K1
f
Round function
L1: 00000000111111101001110111010000
Prev. Step
Auto
Next Step
Skip Round
Intro
Data
PC 1
Key Schedule
IP
DES
FP
Summary
Round
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
54%
DES Visualization

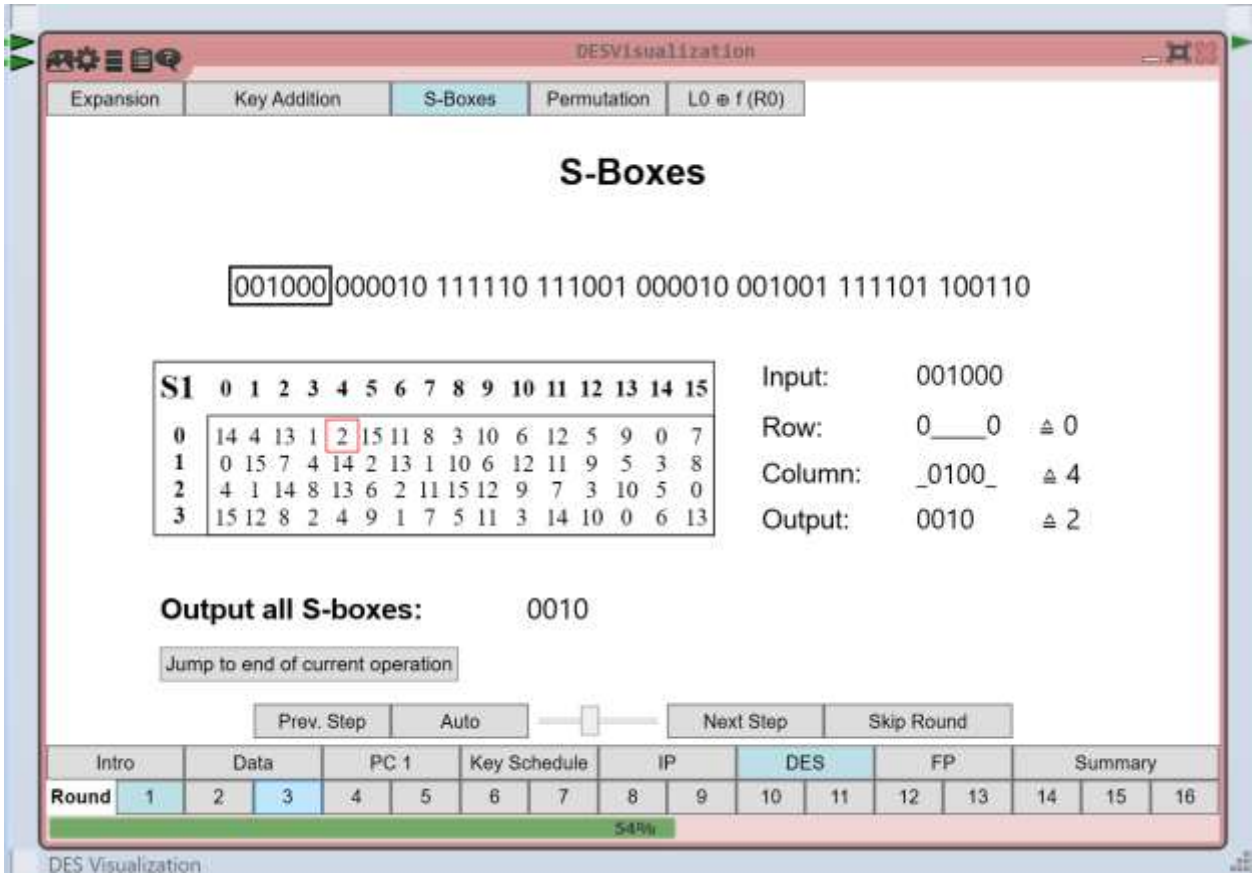
2. Nửa phải được mở rộng thành 48 bit , sau đó đi qua phép XOR kèm kết hợp với khoá K1 (48 bit)



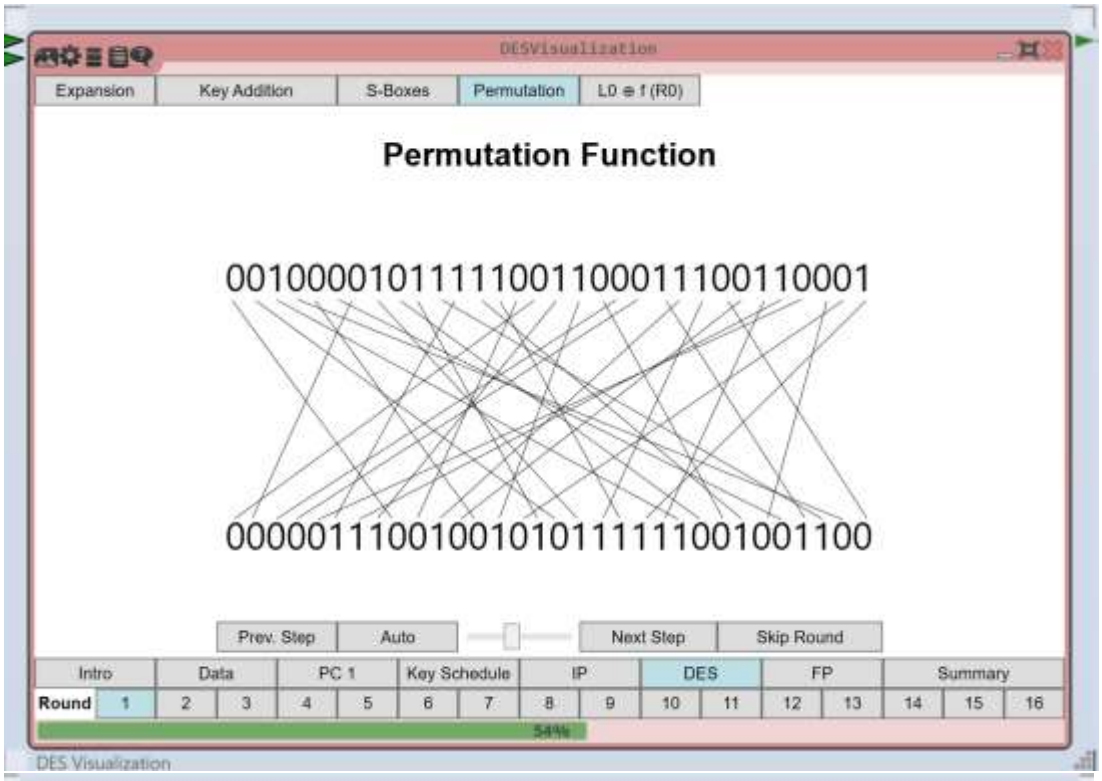
- ở đây sẽ tạo ra giá trị sau phép XOR và kết quả này sẽ đi vào " Thế S-Box "

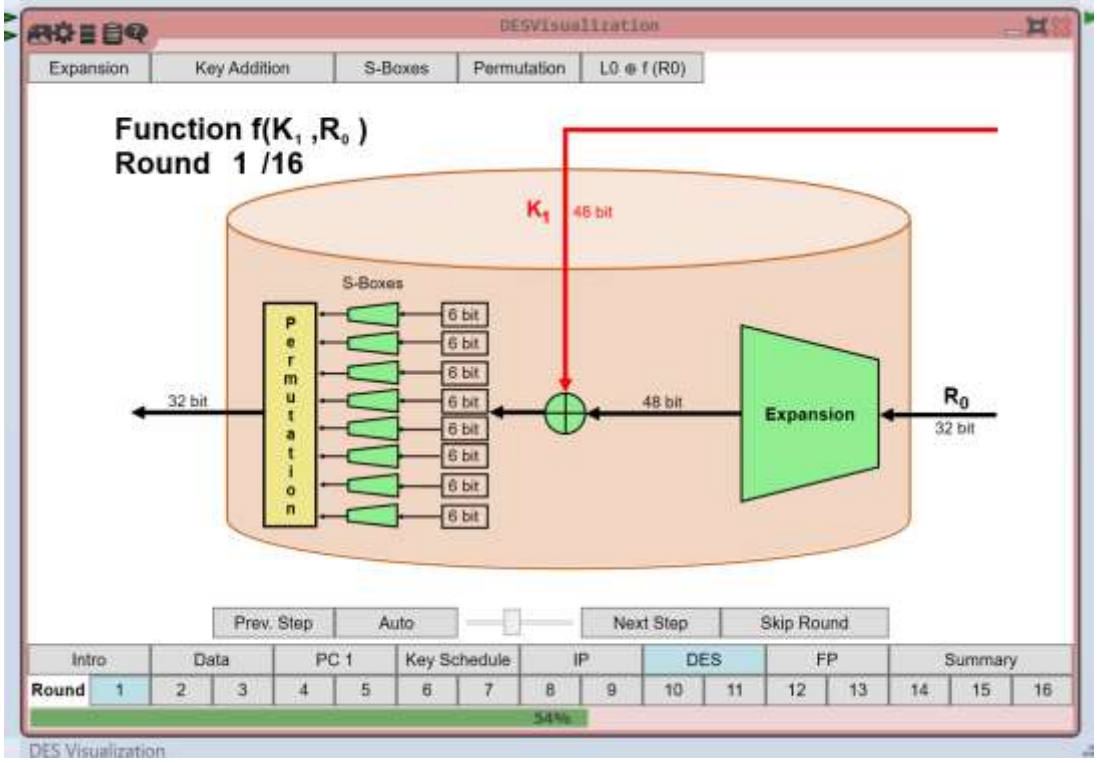


3. Khối dữ liệu sau phép XOR sẽ trải qua thay thế trên S-Box

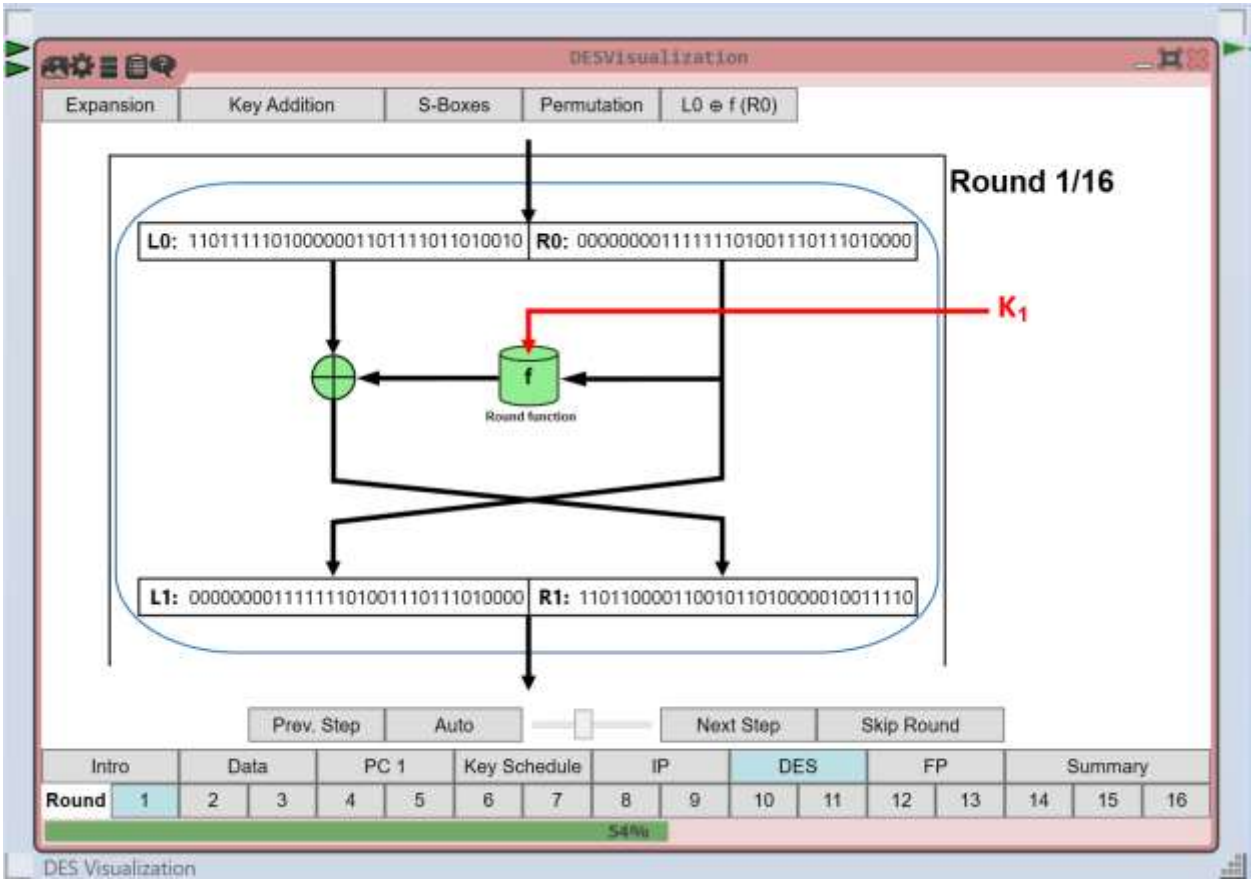


- Giải thích :
  + Dữ liệu được chia thành các cụm 6 bit
  + Bit đầu và Bit cuối của cụm sẽ cho biết vị trí của hàng
  + 4 Bit giữa sẽ cho biết vị trí của cột
  + Có được Ouput cho vòng 1 của S-Box là 2
- Tương tự như vậy, nó sẽ đi qua 8 vòng thay thế trong S-Box
- Sau khi thu được kết quả của 8 vòng S-Box , nó sẽ đi qua hoán vị P và cho ra 32 bit đầu ra

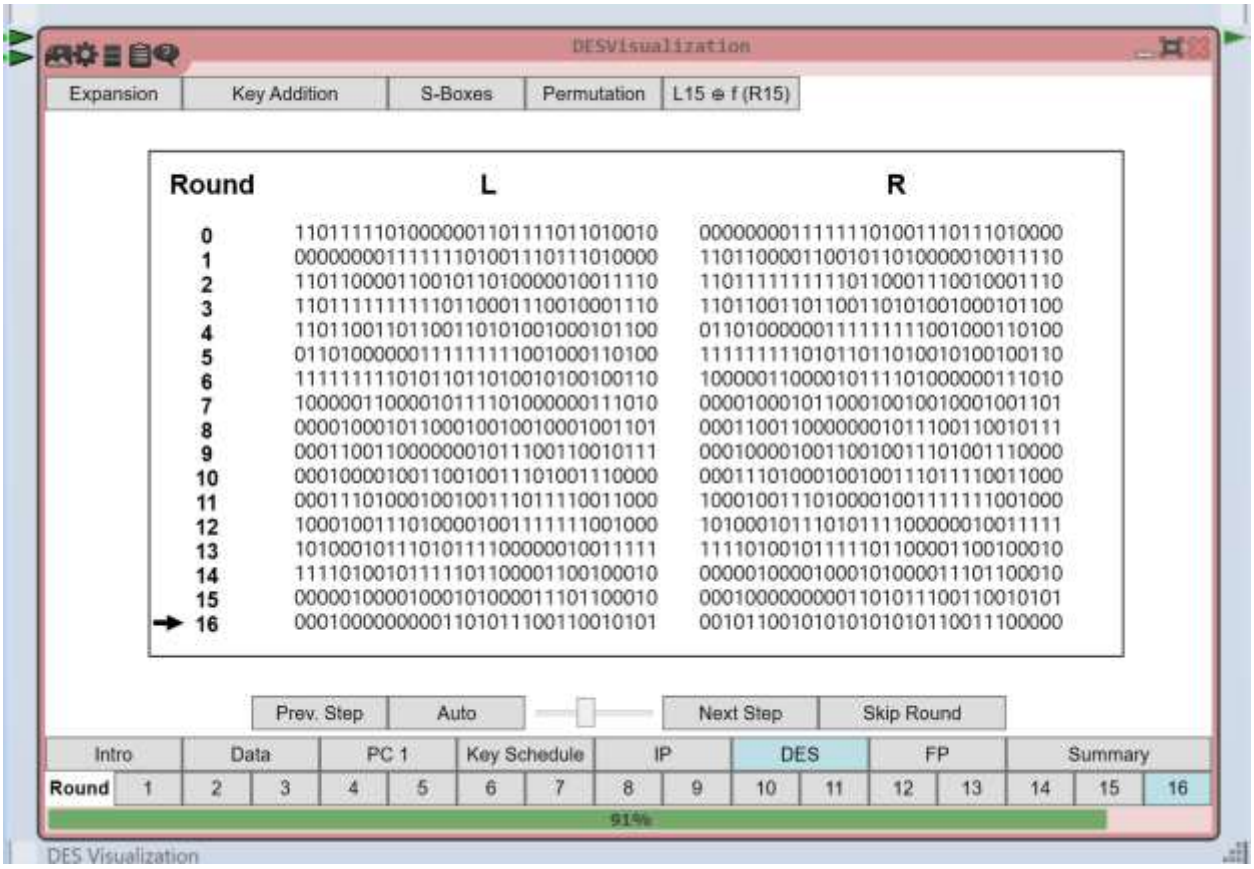

DESVisualization
Expansion
Key Addition
S-Boxes
Permutation
L0 ⊕ f (R0)
Permutation Function
00100001011111001100011100110001
00000111001001010111111001001100
Prev. Step
Auto
Next Step
Skip Round
Intro
Data
PC 1
Key Schedule
IP
DES
FP
Summary
Round 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
54%
DES Visualization


DESVisualization
Expansion
Key Addition
S-Boxes
Permutation
L0 ⊕ f (R0)
Function f(K1 ,R0)
Round 1 /16
K1
48 bit
S-Boxes
Permutation
6 bit
32 bit
48 bit
Expansion
R0
32 bit
Prev. Step
Auto
Next Step
Skip Round
Intro
Data
PC 1
Key Schedule
IP
DES
FP
Summary
Round 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
54%
DES Visualization

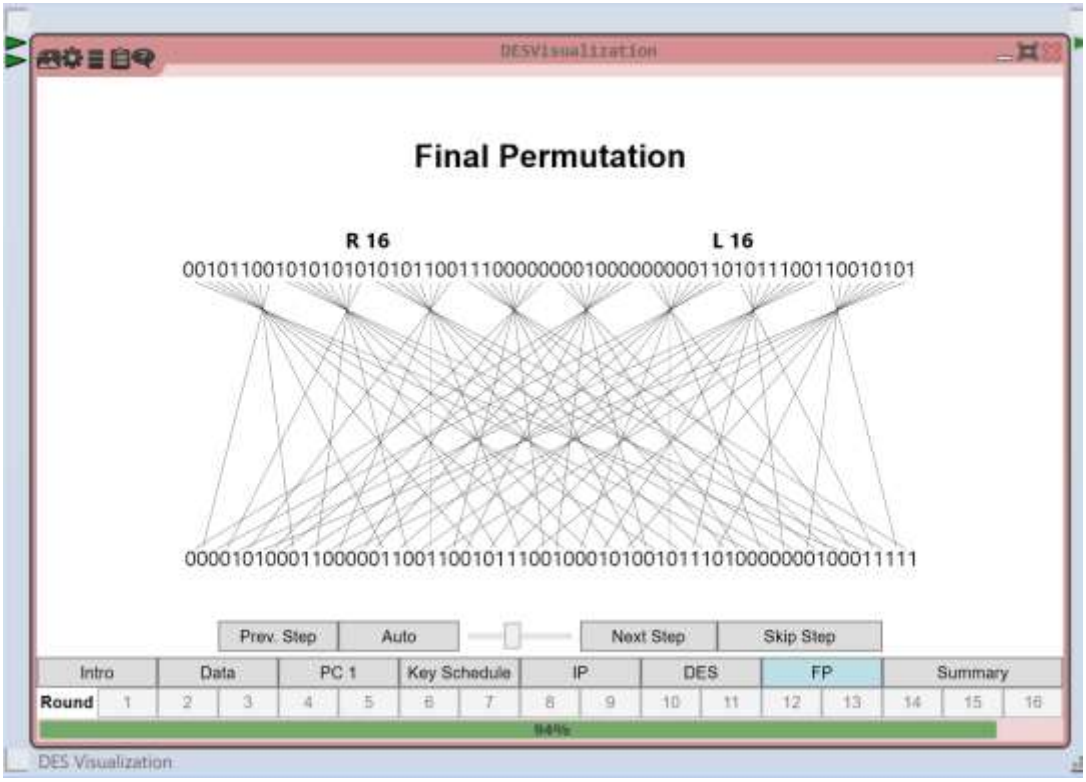
- Kết quả sau khi trải qua hàm Feistel sẽ đi tiếp tục được XOR với nửa trái ban đầu để tạo ra giá trị mới cho nửa phải ở dưới (R1)



- Và tiếp tục như vậy cho đến hết 16 vòng

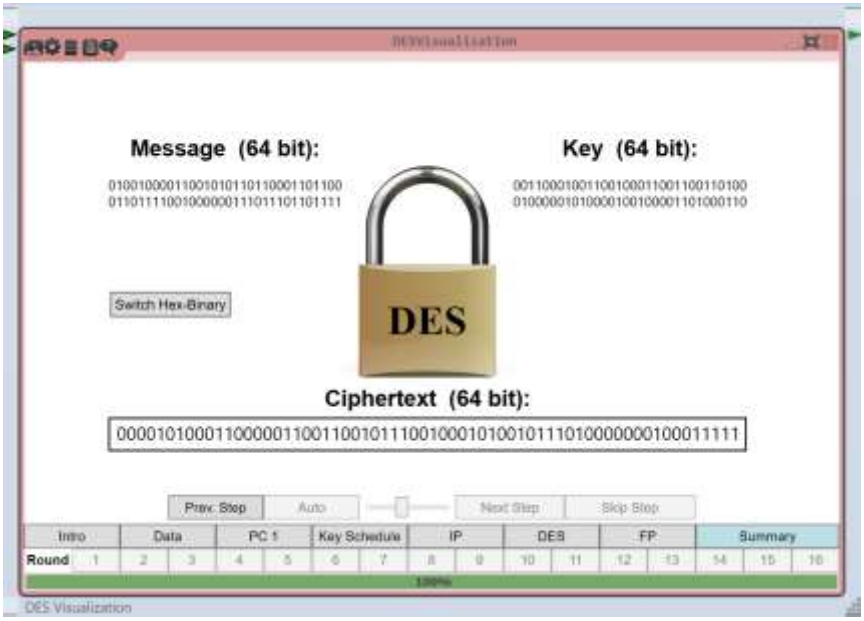


- Sau đó 2 nửa được ghép lại với nhau và đi qua hoán vị cuối cùng (FP)



- Ta thu được Output :

  -Ở dạng binary



- Ở dạng Hexadecimal