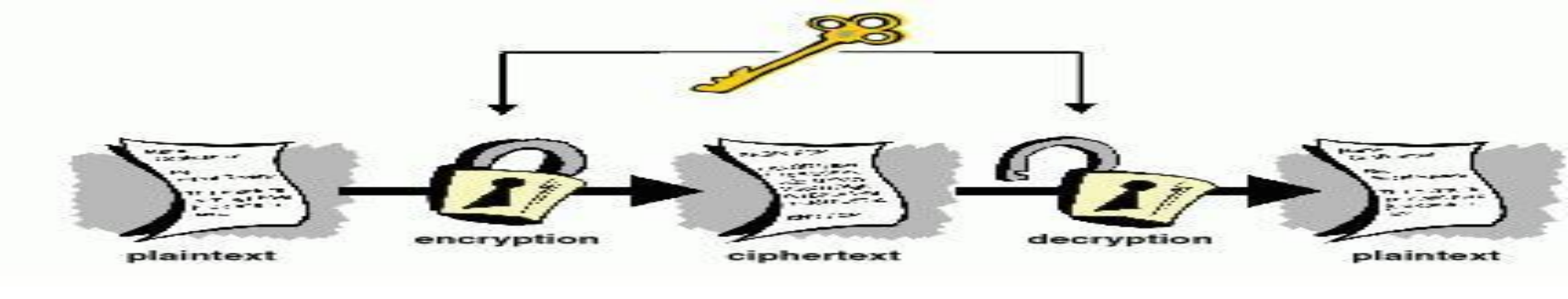
Data Encryption Standard (DES) Implemented in FPGA

By Dinh Pham
216-41-501
An ECE641 Project
Instructor: Mustafa Torun

What is DES?



- DES is a cryptography algorithm that operates on a 64 bit block of data and outputs an encrypted block of data(cipher) also 64 bits large using a key that is again 64 bits.

Image taken from: http://saturn.ece.ndsu.nodak.edu/ecewiki/index.php/Group_SD0726

DES at a glance

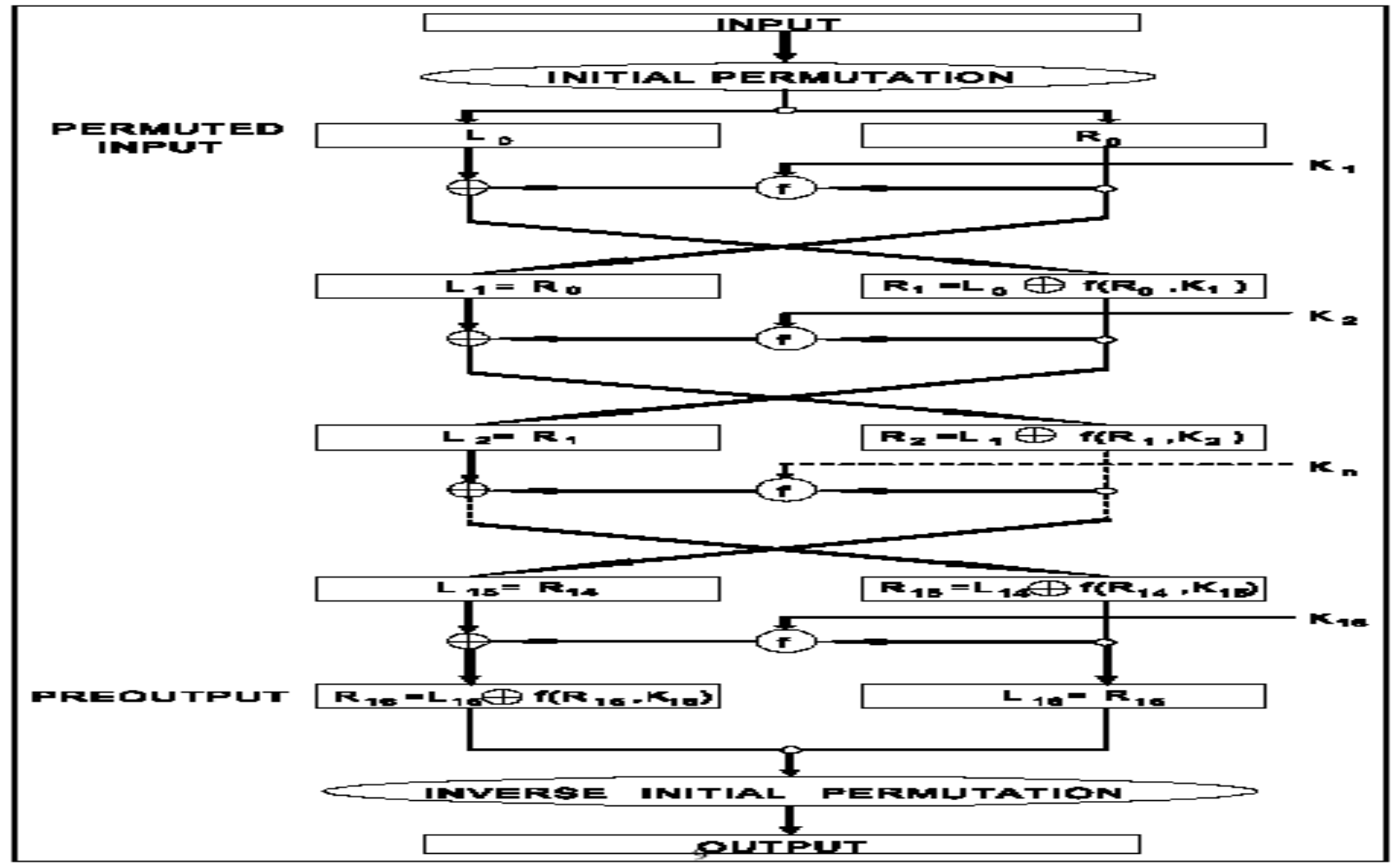


Image taken from: http://www.cerberussystems.com/INFOSEC/stds/fip46-3.htm

Bit Permutations

- One of the biggest components to the DES algorithm are bit permutations, which rearranges the order of bits. Some permutations increases or decreases the bit length.
- The 58th bit of the input becomes the 1st bit of the output of IP., 50th bit becomes 2nd bit of the output, etc. Bits are read from left to right(1st bit=MSB, 64th bit=LSB)

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

Key Scheduling

- The Key Scheduling component generates 16 48-bit subkeys from the initial 64 bit key using bit shifts. These subkeys are then used in the Fiestel function.
- Due to the permutation of the initial key, every 8th bit of the key is unused. This essentially makes the “effective key” 56 bits.

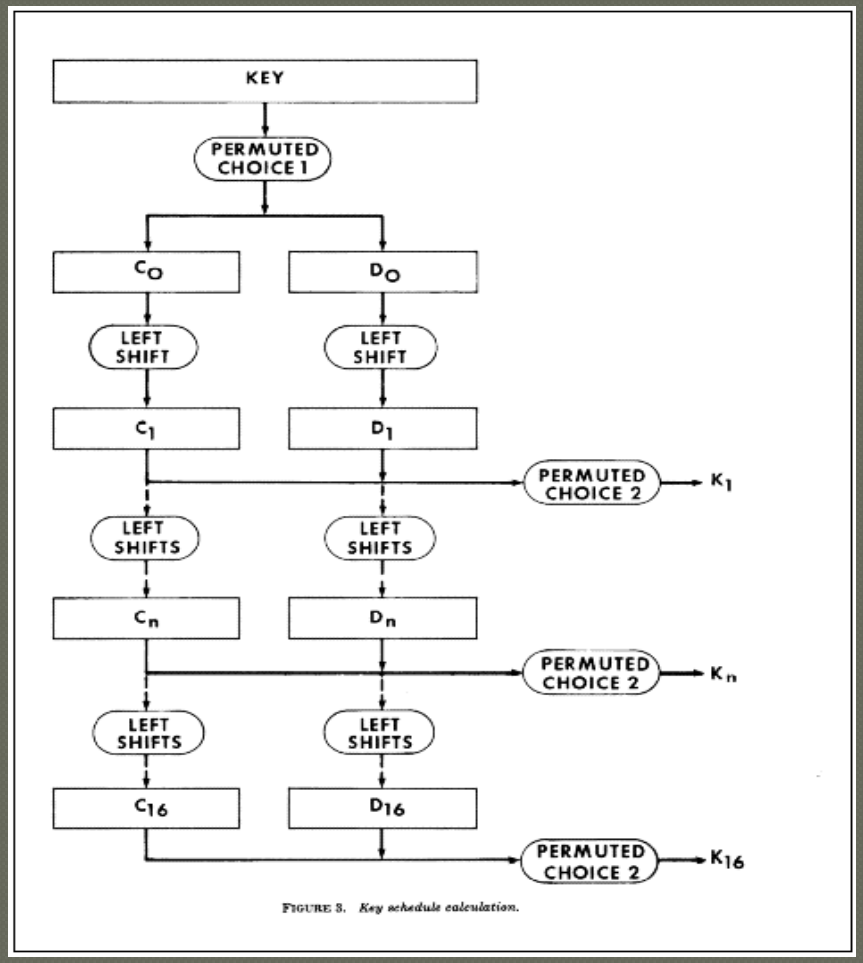


FIGURE 8. *Key schedule calculation.*

Image taken from: http://www.cerberussystems.com/INFOSEC/stds/fip46-3.htm

The Feistel Function

- The feistel function takes a 32 bit input, permutes it to 48 bits and XORs the result with a generated subkey.
- The result is then split into 8 six-bit blocks and transformed into 8 four-bit blocks using the S(substitution)-boxes. This transformation is NOT a typical permutation.

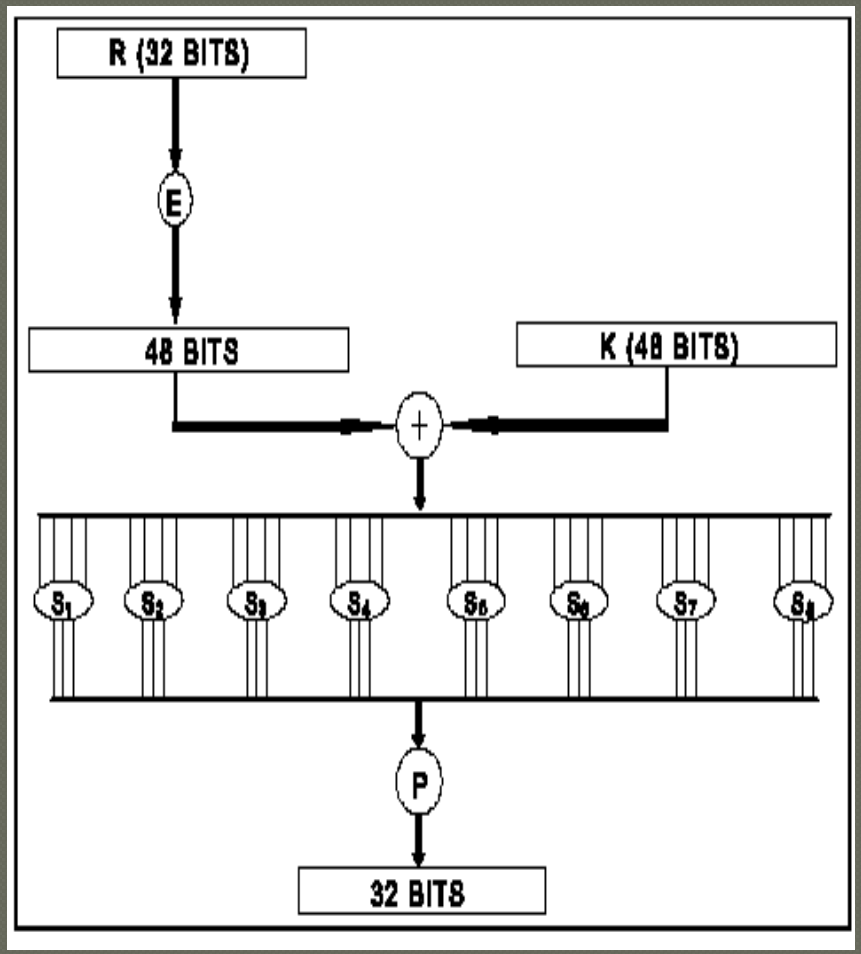


Image taken from: http://www.cerberussystems.com/INFOSEC/stds/fip46-3.htm

S-boxes

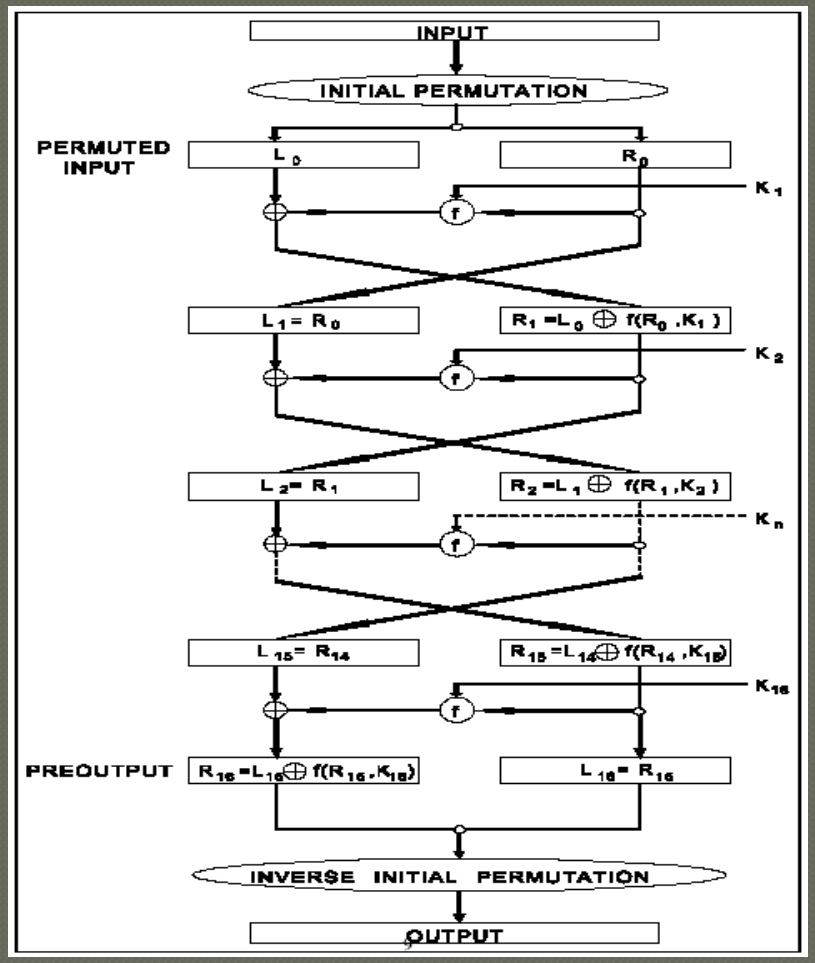
S1

Row No.	Column Number															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

- S-boxes take a 6 bit input and uses the bits in a look up table. The first(MSB) and last (LSB) bits determine the Row Number and the Middle four bits determine the Column Number.
- Example: “011011”, the first and last bits are “01” which corresponds to Row 1. The middle bits “1101” corresponds to Column 13. The output of this S-box would then be 5.
- There are 8 S-boxes each with a different look up table.

Putting together the pieces

- The input message is permutated and split into 32 bit halves, L0 and R0. The R1 is the output of the Feistel function XOR'd with L0. L1 is R(n-1) which would be R0. This process is repeated 16 times(hence the 16 subkeys).
- R16 and L16 are then concatenated and an inverse permutation is performed to generate the cipher text.



Going backwards: Decryption

- One of the perks of DES is that the encryption process is similar to the encryption process. The difference is that the order of the subkeys that are used is reversed. So K16 is applied to R0 in the first Feistel block, K15 is applied to R1 in the second, etc.

Expected Results

- For testing purposes, the following M and Key were chosen:
- M=0000000100100011010001010110011110001001101010101111001 10111101111 or 0123 4567 89AB CDEF in Hex.
- Key=00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001 or 1334 5779 9BBC DFF1 in Hex.
- Given those M and Key, the expected Ciphertext is 10000101 11101000 00010011 01010100 00001111 00001010 10110100 00000101 or 85E8 1354 0F0A B405 in Hex.

Test and expected values taken from http://www.orlingrabbe.com/des.htm

References

- **The DES Algorithm Illustrated** http://www.orlingrabbe.com/des.htm
- **Wikipedia: Data Encryption Standard** http://en.wikipedia.org/wiki/Data_Encryption_Standard
- **Federal Information Processing Standards Publication: Data Encryption Standard** http://www.cerberussystems.com/INFOSEC/stds/fip46-3.htm
- **VHDL Tutorial** http://www.seas.upenn.edu/~ese171/vhdl/vhdl_primer.html