FAST NATIONAL UNIVERSITY
School of Computing
Spring 2021

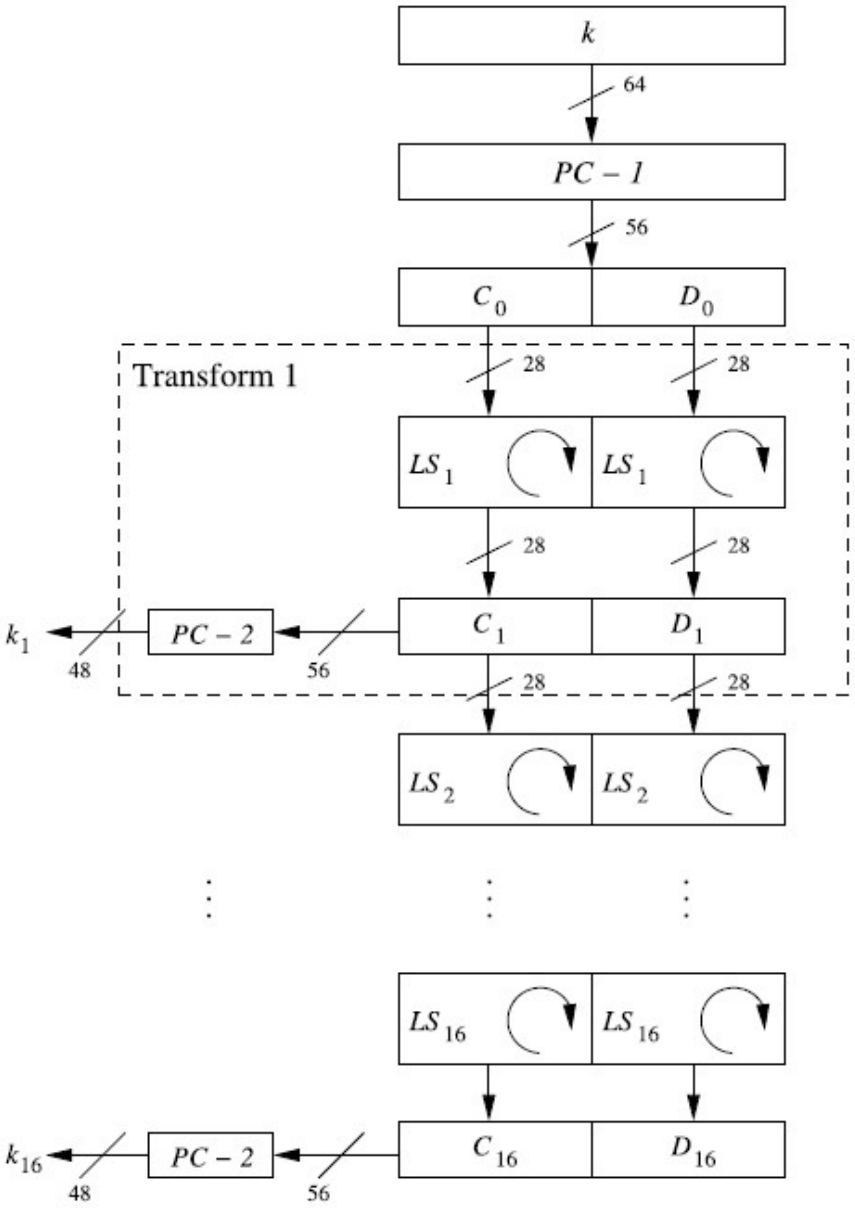
Course Title:	Computer Organization and Assembly Language
Task:	Assignment #7
Weightage:	6%

Q1. In this assignment, you can work in group of two students. You have to implement the "Key schedule" function, which derives 16 round keys k_i, each consisting of 48 bits, from the original key. Another term for round key is subkey. The 64-bit key is first reduced to 56 bits by ignoring every eighth bit using the PC-1 permutation as follows:

Initial key permutation $PC-1$

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

The resulting 56-bit key is split into two halves and the key schedule starts as shown in figure below.



Key schedule for DES encryption

The two 28-bit halves are cyclically shifted, i.e. rotated left by one or two bit positions depending on the round *i* according to the following rules:

- In round *i* = 1, 2, 9, 16, the two halves are rotated left by one bit.
- In the other rounds, the two halves are rotated left by two bits.

Note that the rotations only take place within either the left or the right half. To derive the 48-bit round keys, the two halves are permuted bitwise again with PC-2 and it is shown in figure below:

Round key permutation $PC-2$

$PC-2$							
14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

In this assignment, your task is to implement the key schedule function in Assembly language as a subroutine and share its code.