1) Using T-FF’s design and implement the register transfer system that has following operation features:

C1: \[ A \leftarrow B + 1 \]
C2: \[ A \leftarrow A' + 1 \] (2’s complement)
C3: \[ A_{n-1} \leftarrow 0, \quad A_i \leftarrow A_{i+1} \] (shift right)
C4: \[ A \leftarrow A - 1 \] (decrement)

\( A \) and \( B \) are 5-bit each

2) A digital system has 16 registers, each with 32 bits. It is necessary to provide parallel data transfer from each register to each other register.

a) How many lines are needed for direct parallel transfer?

b) How many lines are needed for transfer along a common bus?

c) If the registers form a scratch-pad memory, how is information transferred from one register to another?

Let the register in the memory be designated by \( R_0 \) to \( R_{15} \). List the sequence of micro operations for a transfer of the content of \( R_6 \) into \( R_{13} \).

3) Draw the logic diagram of a 4 bit register with clocked JK flip-flops having control inputs for the increment, complement, and parallel transfer micro-operations. Show how the 2’s complement can be implemented in this register.
4) Suppose a computer with the following characteristics if given:
   word length: 32 bits
   floating point number representation:
     - exponent: field size: 7 bits
       base (radix): 16
       representation: excess-64 (i.e. 64-biased)
     - mantissa: field size: 24 bits
       radix point: before the first digit
     - sign: field size: 1 bit
       2's complement

<table>
<thead>
<tr>
<th>1</th>
<th>7</th>
<th>24</th>
</tr>
</thead>
<tbody>
<tr>
<td>sign</td>
<td>exponent</td>
<td>mantissa</td>
</tr>
</tbody>
</table>

a) Show how the following number should be stored in a word within this computer:

\[-(3B1.AA10)_{16}\]

b) What are value of the smallest positive number and the most negative number which can be represented?

5) Using JK FF’s design and implement the register transfer system that has the following operating features: (registers A and B are each 4 bits)

T1: A ← A’ + 1
T2: A ← A – 1
T3: A ← B + 1
T4: An-1 ← 0, Ai ← Ai+1 shift right
T5: A ← A ⊕ B