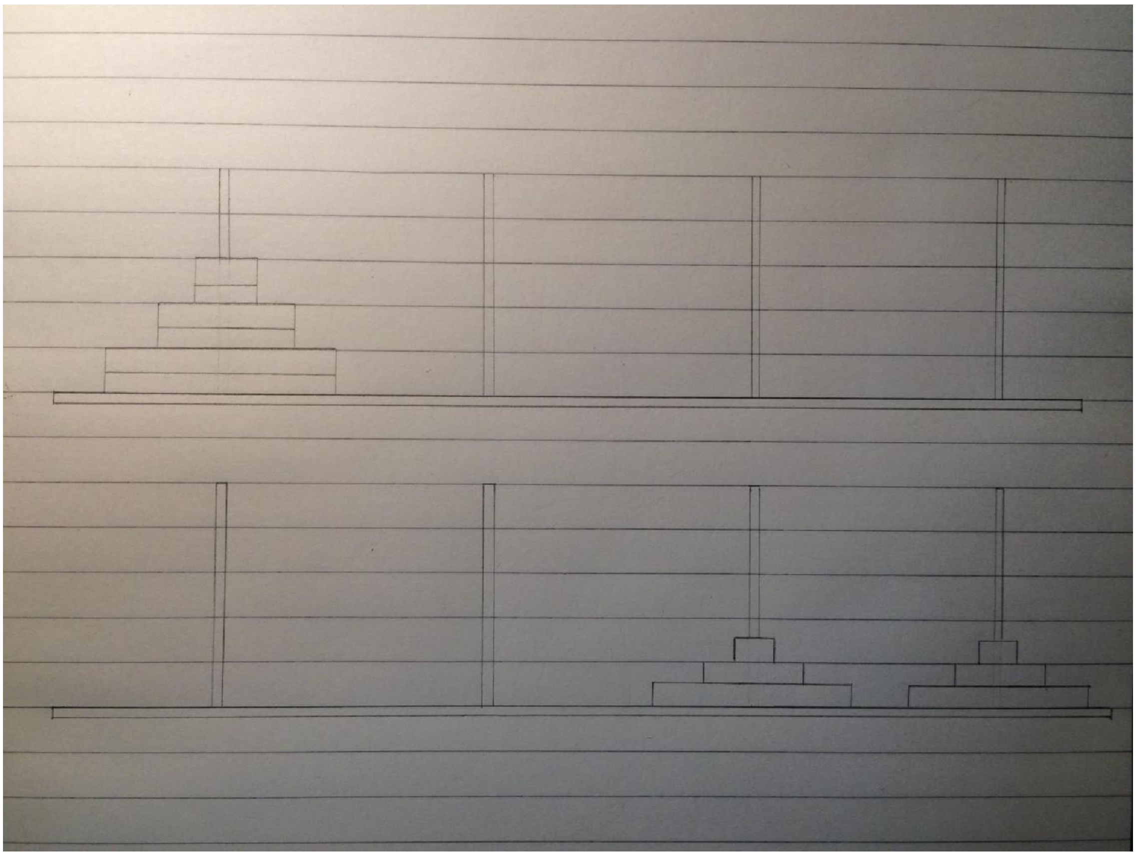
PROJECT ABSTRACT

DESIGN OF DIGITAL SYSTEMS (CO-202)

[TEAM MEMBERS: AMAL BYJU(ROLL NO: 16CO205), EDWIN THOMAS (ROLL NO: 16CO218)]

THE SUBTOWER PROBLEM

OBJECTIVE: This challenge begins with 6 disks on the first peg of four pegs (3 pairs of disks of equal size) stacked up in decreasing order of size. You are supposed to transfer the disks (one at a time) such that you get two towers of equal size at pegs 3 and 4.



APPARATUS: D-flip flop(IC: 7474), T-flip flop(IC:7476), and gate(7408), nand gate(IC:7400), or gate(IC:7432), 7-segment display, connecting wires, clock.

PROJECT DESCRIPTION:

The objective of this puzzle is to move the entire stack to pegs 3 and 4, making 2 towers of equal size at these pegs. The following rules should be followed:

1. Only one disk can be moved at a time.
2. Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack.
3. No disk may be placed on top of a smaller disk.

This project demonstrates the implementation of an iterative algorithm for the Tower of Hanoi puzzle which can be of use for this problem.

The Subtower Algorithm

SUBTOWER_ALGORITHM

1)FromPeg, ToPeg are considered to be arrays [1 ... n] with possible values from one to four.

```
Initial States
  FromPeg <-{1, 1, 1, 1, ... 1}
  if n is odd:
     ToPeg <-{3, 2, 3, 2, ... 3}
  else
     ToPeg <-{2, 3, 2, 3, ... 3}

for i <- 1 to 2^(n+1) - 2:
        begin
        Find the largest d (1 <= d <= n) such that i mod 2^(d)-1 =
        0 Move disk d from peg FromPeg[d] to peg ToPeg[d];
        label:
        case[FromPeg[d], ToPeg[d]] of

          [3,2] FromPeg[d] <- 4 ToPeg[d] <- 2;
               goto label;
          [1,2] FromPeg[d] <- 2 ToPeg[d] <- 3;
          [4,1] FromPeg[d] <- 1 ToPeg[d] <- 2;
          [2,1] FromPeg[d] <- 1 ToPeg[d] <- 3;
          [2,4] FromPeg[d] <- 3 ToPeg[d] <- 1;
          [1,3] FromPeg[d] <- 1 ToPeg[d] <- 4;
               goto label;
          [1,4] FromPeg[d] <- 3 ToPeg[d] <- 2;
          [3,1] FromPeg[d] <- 4 ToPeg[d] <- 1;
               goto label;
          [4,2] FromPeg[d] <- 2 ToPeg[d] <- 1;
          [2,3] FromPeg[d] <- 2 ToPeg[d] <- 4;
               goto label;


       end; {case}
       end; {for}
       end; {procedure}
```

IMPLEMENTATION OF DIGITAL DESIGN FOR THE SUBTOWER PROBLEM

STEP 1: The 6 disks are divided into 3 categories on the basis of their sizes (category 1 for the smallest). The row index number of the initialised array shows the size of the disk to be moved at each step. For the above purpose we implement a four bit counter made purely of synchronous T-flip flop elements. The four bit counter is used to increment the i variable at each iteration and correspondingly generate the row index number(rowd) analogous to the d value. By observation the row number to be altered corresponds to the position of the first nonzero bit output from the counter seen from LSB.

STEP 2: Each of the four outputs of the synchronous counter is connected to four memory units giving them their enable signal. The memory unit each consists of four d flip flops. With reference to the array Frompeg and Topeg the d flip-flops in each memory unit is initialized. The output to these four memory units is a tri-state bus whose enable is the row d signal. The current value of the d flip flop in every iteration is the current output the tri-state bus. Moreover, a combinational circuit finds the next state of the four d flip flops based on the current values.

NOTE:

1. The 7-segment display shows the output at both the positive and negative edges of the clock cycle.
2. The output of the tristate bus at every iteration gives a four-bit output that corresponds to each move to be done in sequential order to solve the Tower of Hanoi problem.(We move the top most disk from Frompeg to Topeg values).

REFERENCES

1) https://forums.xilinx.com/t5/Simulation-and-Verification/doubts

2) https://www.allaboutcircuits.com/

3) http://www.electrobucket.com/