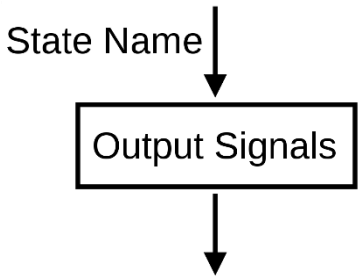
ASM diagram for *Rope Simulation* using *Verlet Integration* and *Jacobsen's Method*

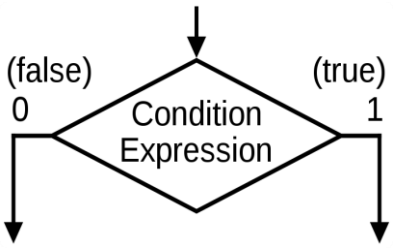
In this report, the simulation steps of rope will be explained with help of the Algorithm state machine.

A quick review of *ASM*:

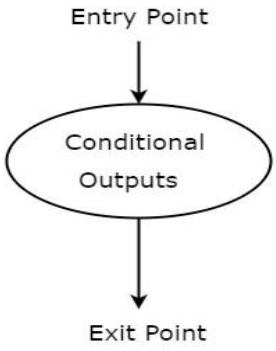
- State box: A rectangle representing the state of the system.



- Decision box: A diamond indicates that the stated condition/expression is to be tested and the exit path is to be chosen accordingly.



- Conditional box: An oval contains outputs depending on both state and input. (Mealy design)



What is our goal?

Our goal is to find a way to simulate a rope movement.

We don’t need to come up with a perfect simulation. we are interested in techniques that give us the most fidelity at a low computational cost.

The approach to the rope simulation problem

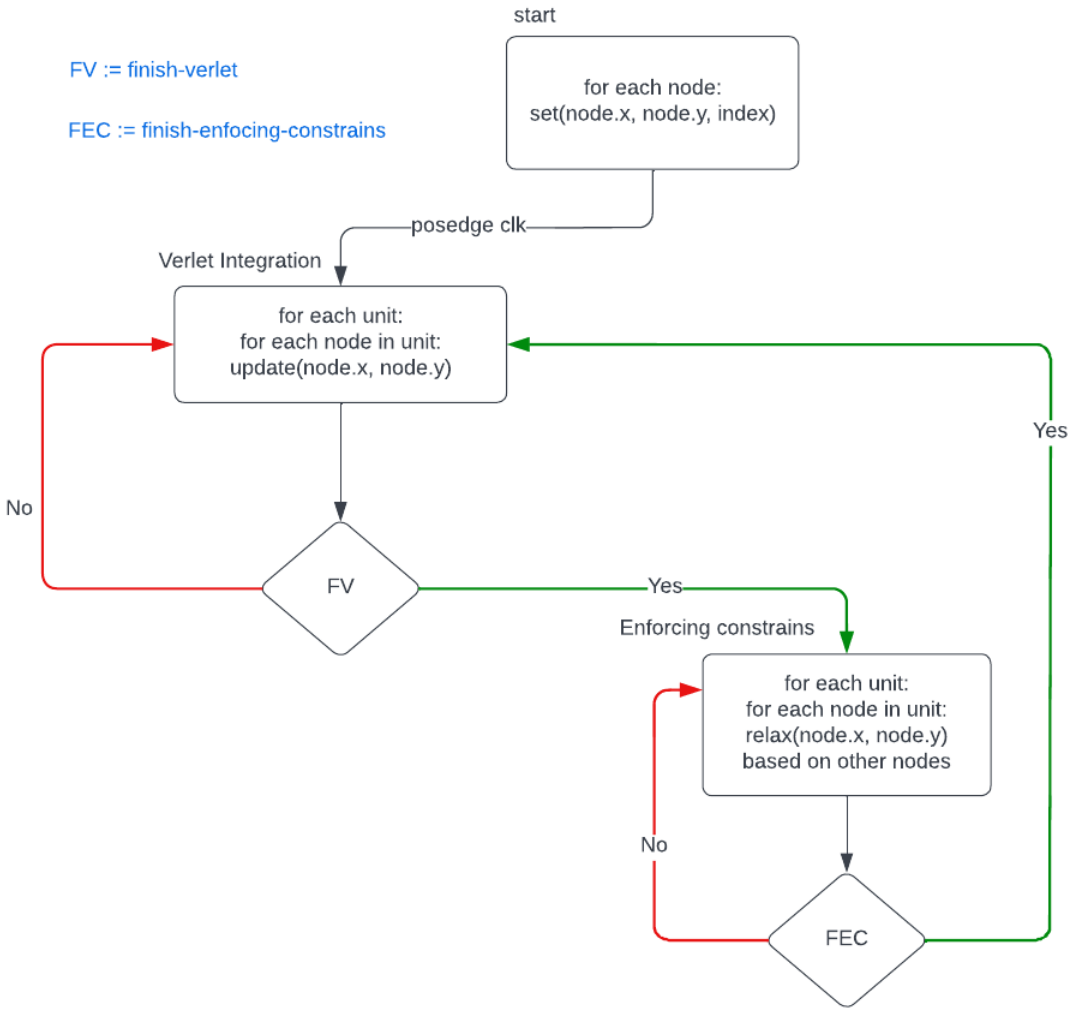
As we mentioned before, our simulation does not need to be perfect therefore we come up with these assumptions:

1. The rope is a collection of finite nodes which are connected with ropes that has no mass.

2. The effect of gravity or external forces, affect each node, and nodes do not affect each other.

3. To get an acceptable result we will relax the last assumption by fixing the distance between every two nodes. In that case, the Tension force effect is considered.

The Big Picture Of The System



As it is shown in the above ASM, the system contains three main parts:

- Start state: creating nodes and set the default position

- Verlet integration block: calculating the next state of each node based on external forces and gravity with

the assumption that nodes don't have any effect on each other. If there are any external forces, the system will affect them on nodes.

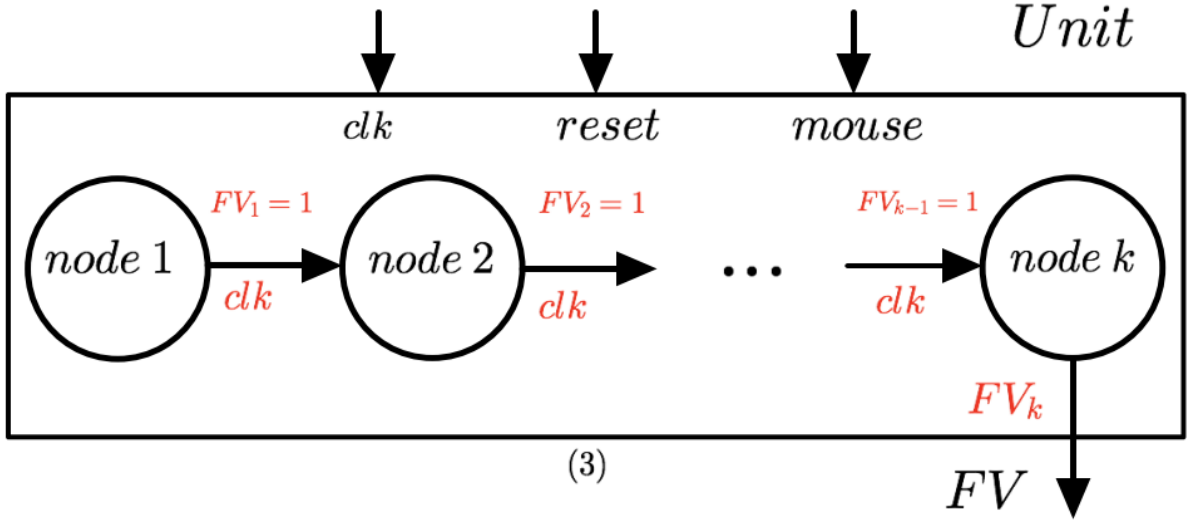
- Enforcing constraint block: based on the third assumption, the position of the node must be relaxed based on the fixed distance that is set for every two nodes.

! We will discuss FV and FEC signals later.

Too Many Nodes Problem

For better results, we can increase the number of nodes however we have few processors therefore each processor must take care of more than one node.

Each unit is a pack of nodes and units can execute some steps parallel. We assign each unit to one processor.



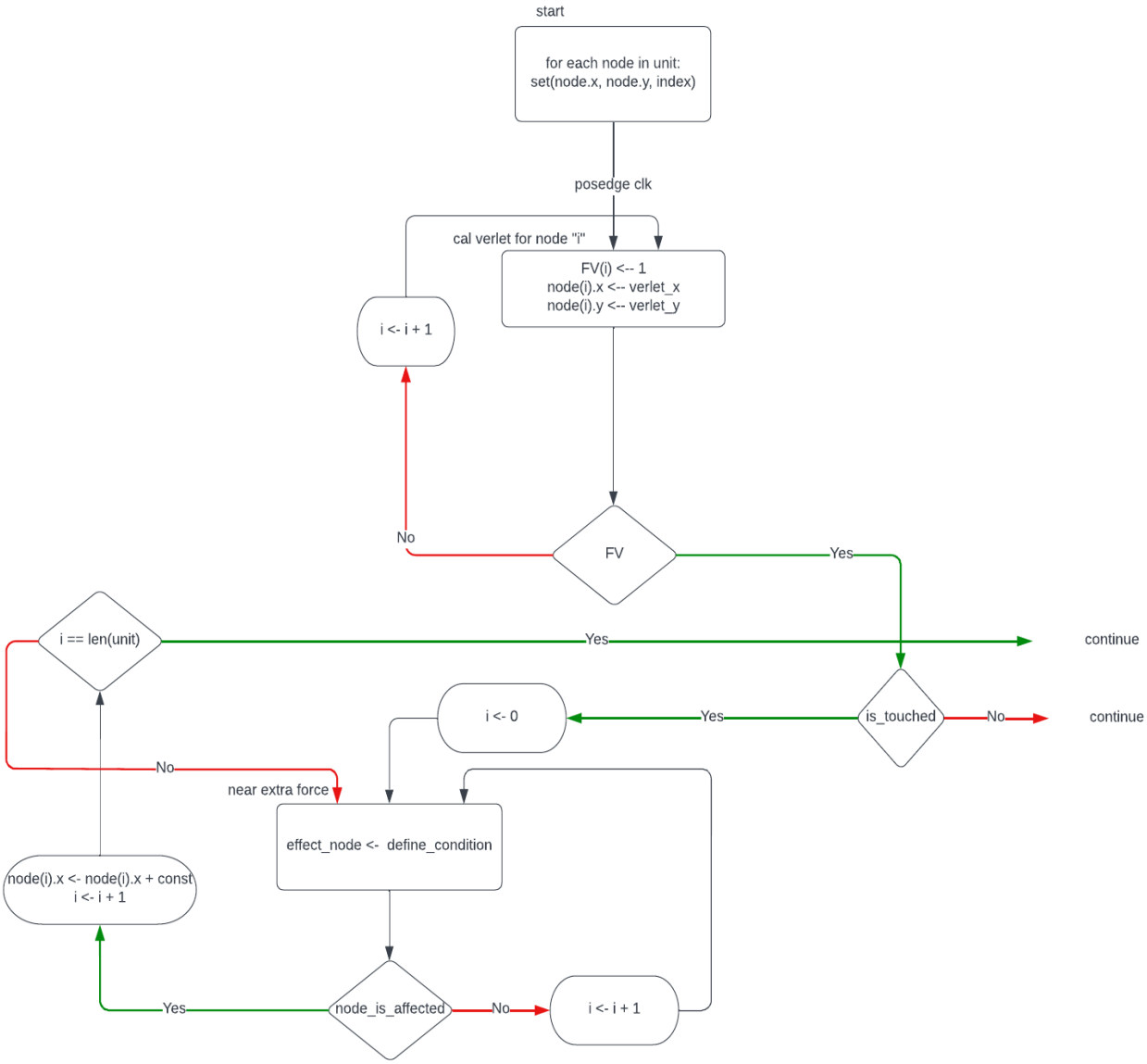
(3)

As shown in the above diagram, the *FV* signal is sent when the Verlet integration calculation of all nodes in the unit is done.

Verlet Calculation Block

We discuss the rule of the Verlet calculation block in previous parts, it’s time to deep scan this block.

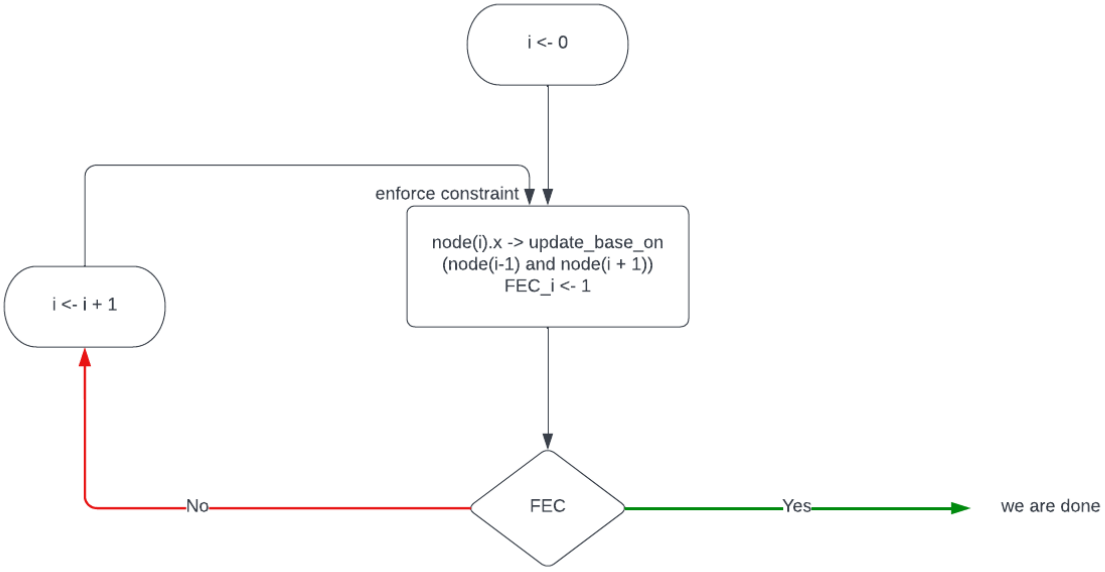
As we mentioned in “Too many nodes problem” we have units that are assigned to each processor therefore checking the process of one unit is enough.



As we can see in ASM, the system iterates over each node and updates the position, if there is any extra force, we will update the nodes near the force by adding a const to their position. (We assume that the direction of the force is parallel to the X-axis.)

Enforcing Constraints Block

To have a better result, the system must consider the tension of the rope somehow. So, the distance between every two nodes is set to be a constant. With this assumption, the position of every node will be relaxed.



As it is shown in ASM, the system relaxes all of the nodes based on the distance between the upper and lower nodes. (FEC signal is the logical and of all FEC_is)

More information about enforcing state is provided in FSM report.