

22c:181 Formal Methods in Software Engineering

Part II

Reactive Systems and the Lustre language

Christoph Stickse
christoph-stickse@uiowa.edu

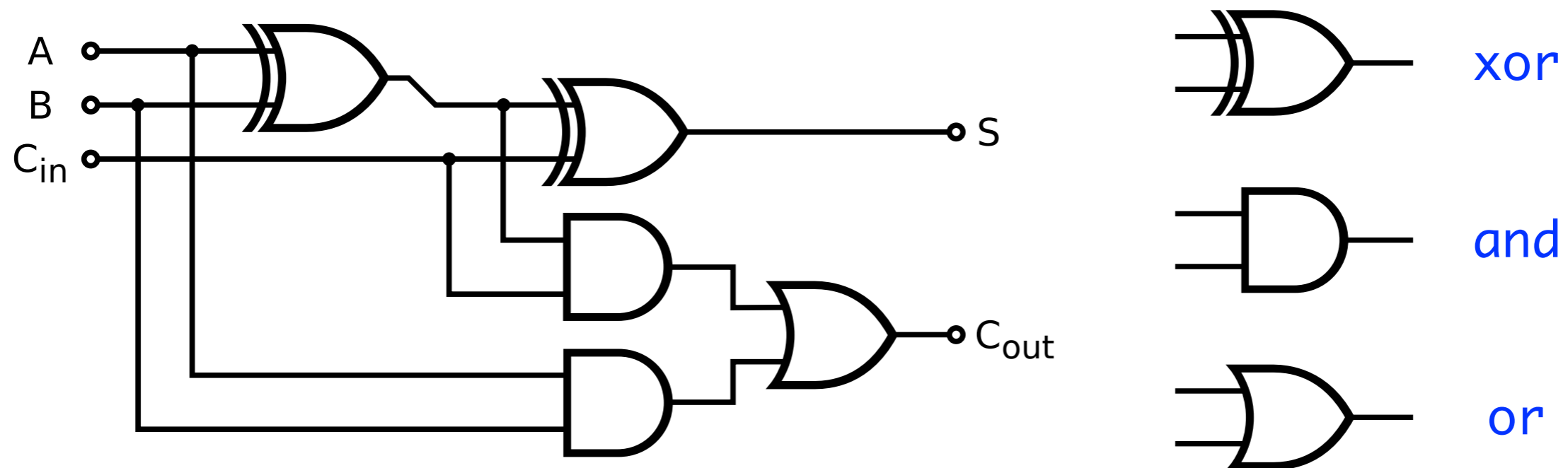
Exercises

1. A node without inputs that outputs $X = 1, 2, 3, 3, 3, \dots$

Use only constants, \rightarrow and **pre**

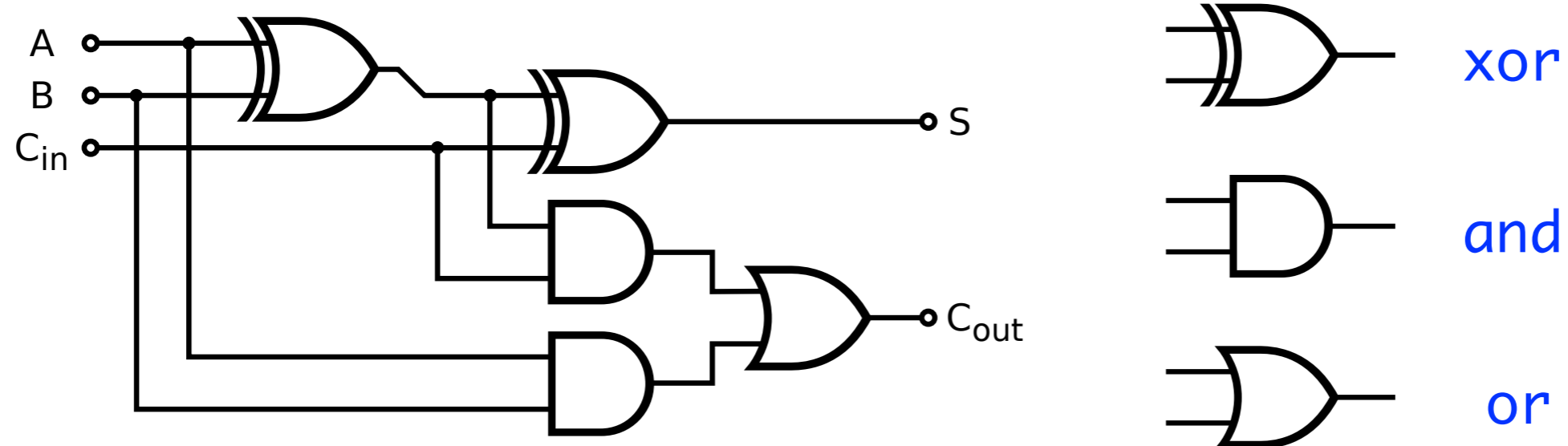
2. A node without inputs that sums up the integers $1, 2, 3, \dots$

3. A node that implements the full adder circuit

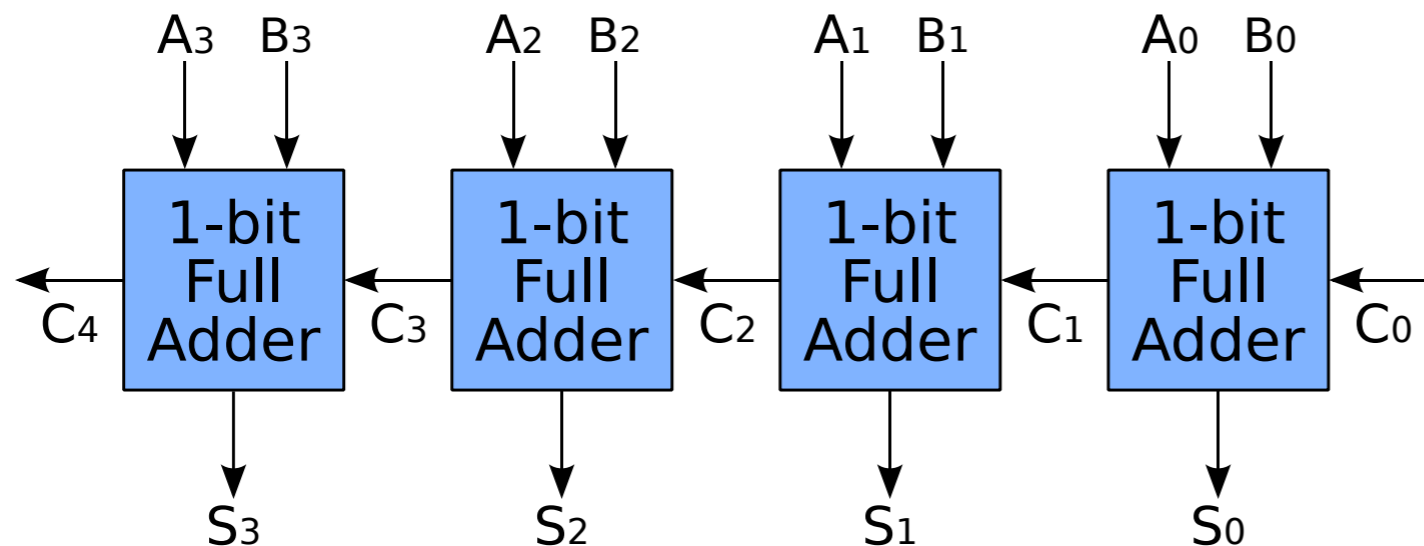


Exercises (2)

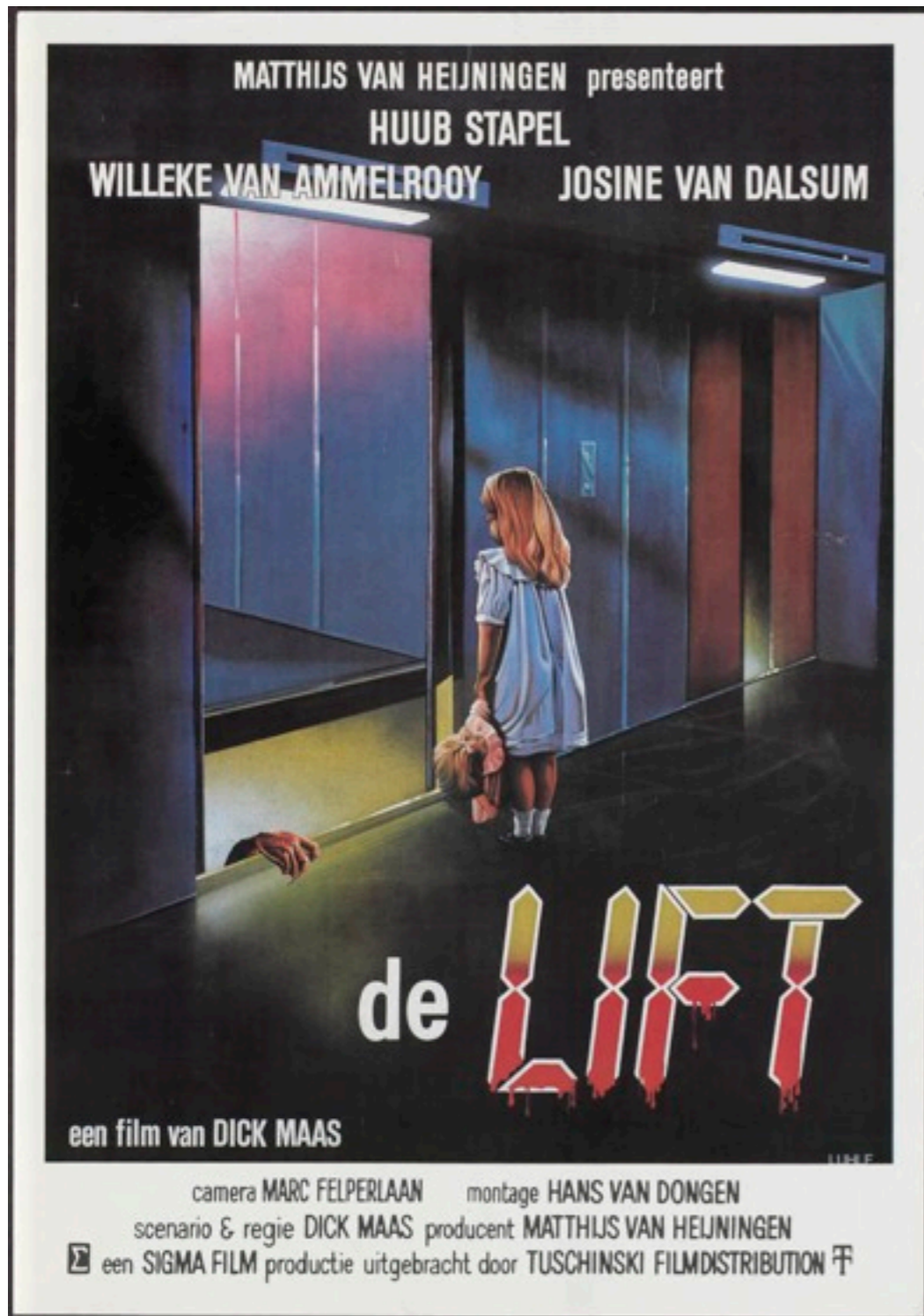
3. A node that implements the full adder circuit



4. A four-bit adder using four full adder circuits

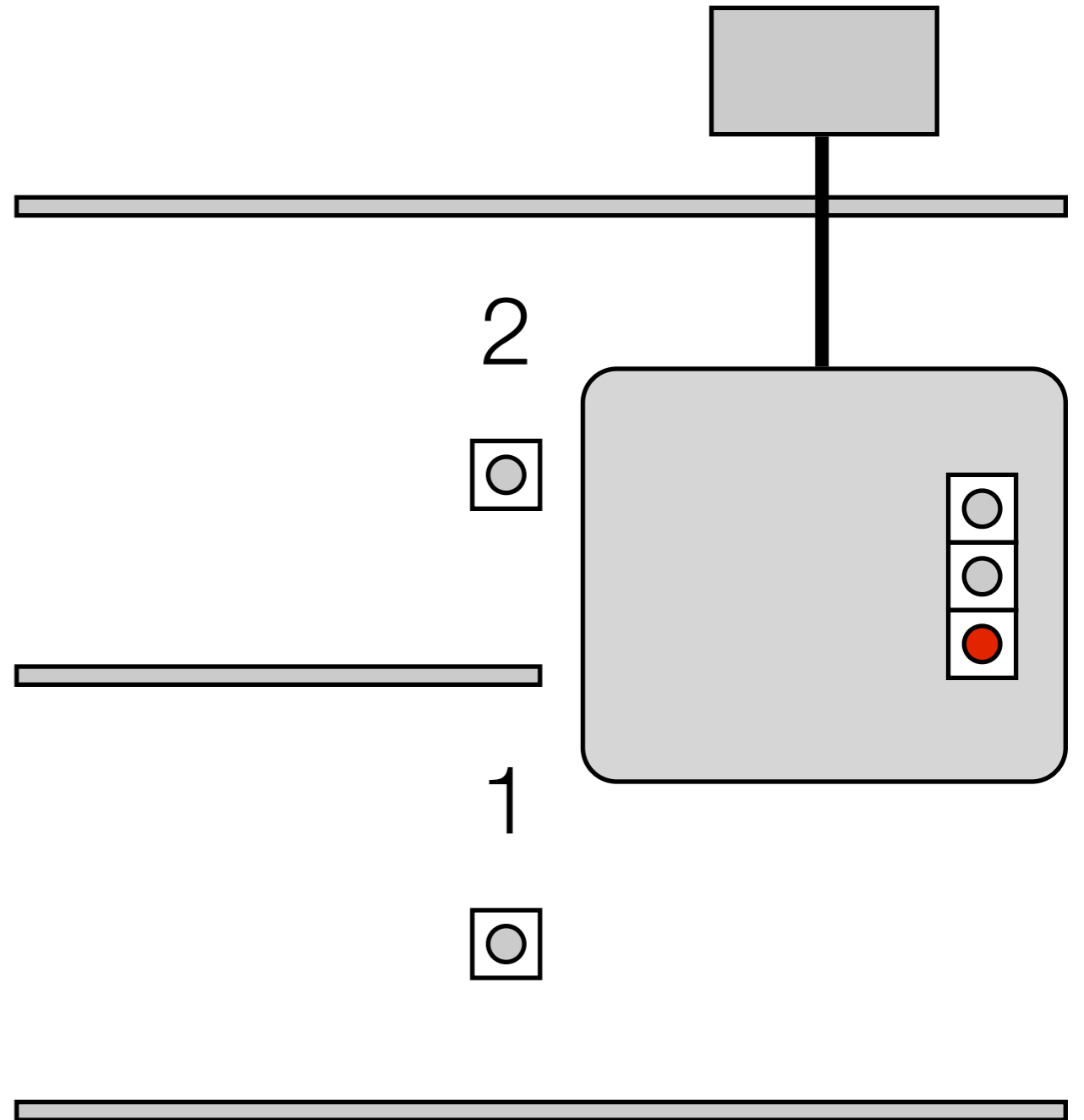


Elevator Controller



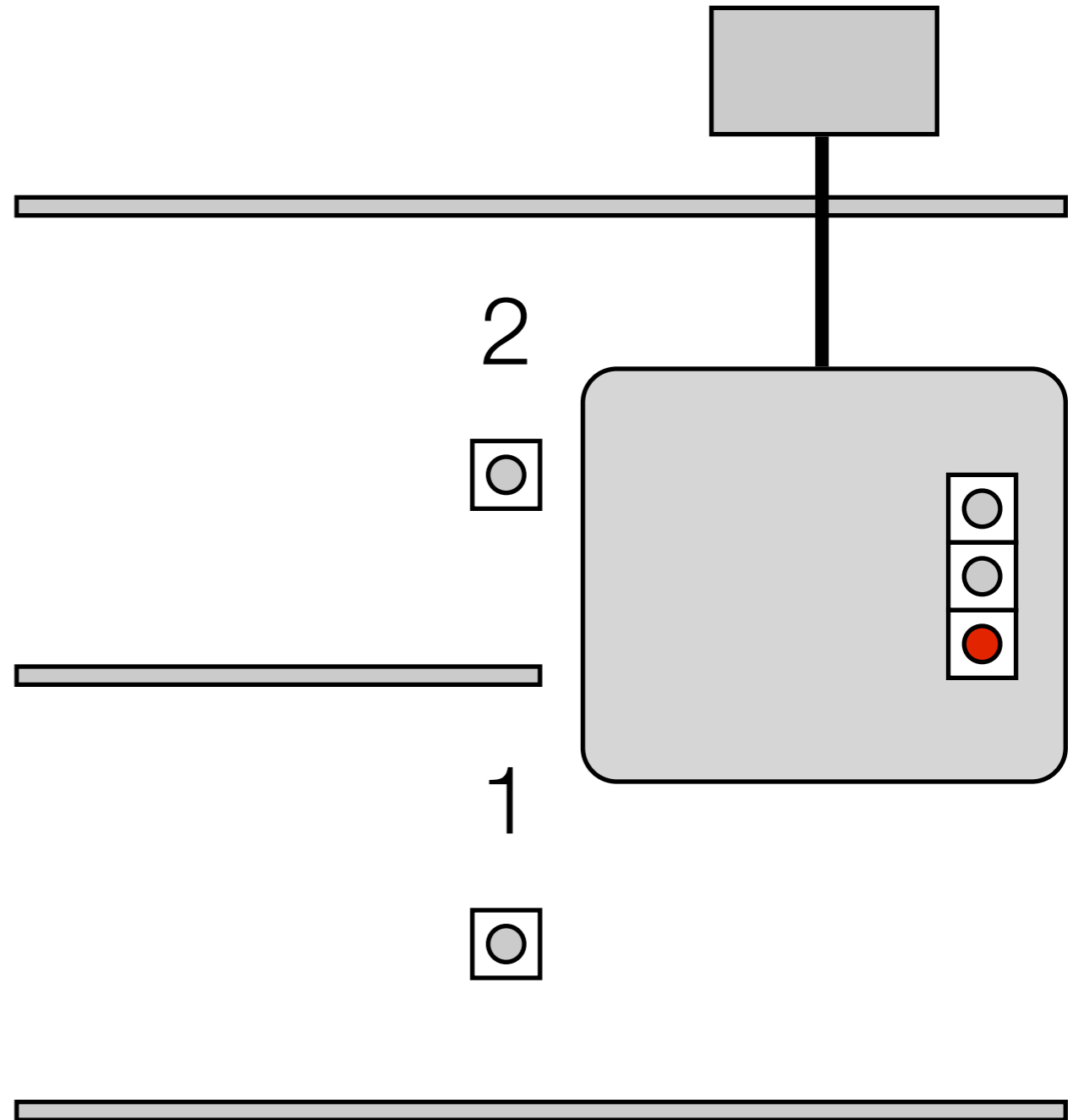
Take the stairs, take the stairs. For God's sake, take the Stairs!!!

Elevator Controller



Elevator Controller

```
node Control( Floor_1, Floor_2, Door_Open,  
              Call_1, Call_2, Stop : bool )  
returns ( Motor_Up, Motor_Down : bool );
```



Elevator Controller

```
node Control( Floor_1, Floor_2, Door_Open,  
              Call_1, Call_2, Stop : bool )  
returns ( Motor_Up, Motor_Down : bool );
```

1. The elevator may only move when the door is closed and the stop button is not pressed.
2. The elevator may not pass the end positions, that is, go through the roof or the floor.
3. A moving elevator halts only if the stop button is pressed, or the door is opened, or the elevator has arrived at the destination floor
4. The elevator must halt before changing direction.
5. The signals sent to the motor may not be contradictory.
6. The elevator moves if it is at a floor and the call button for another floor is pressed, the door is closed and the stop button is not pressed.

