

Design a digital controller that operates the elevator. It takes the input from the user and start operating depending on its state as will be explained below

Entity name should be controller, and the port names should be named as the following (in order), This is used for autograding please don't change names.

Inputs:

1. Clk (1-bit)
2. Rst (1-bit): 1 → **asynchronous** and set all registers to a value of 0.
3. Valid(1-bit): 1 → means the data is valid.
4. Choice (3-bit): this contains the button the user hit (7 floors + ground floor)

Outputs:

- Action (2-bits) : 00 idle (not moving), 01 moving upwards, 10 moving downwards. 11 openDoor.
- Floor (3-bits): current floor

Rst signal: resets all the operations to start from the beginning, and set any internal register to its initial value.

The functionality works as follows, when rst =1 , all outputs and internal states are initialized.

Each cycle the elevator take a decision on one of the four actions to perform based on its internal state and the registered choices. The elevator prefers to keep its moving status rather than changing it i.e. If the elevator was in the 3rd floor and moving upward and both choices 1 and 6 are chosen, the elevator should move upwards first.

The elevator takes a cycle to open the door, it keeps a previous history of its last moving state when it opens the door to continue in the same direction afterwards if there are more floors chosen, check the timing diagram below

Example: X means don't care you don't need to put X value on the data

Cycle#	1	2	3	4	5	6	7	8	9	10	11	12	13
Rst	1	0	0	0	0	0	0	0	0	0	0	0	0
Valid	x	1	0	1	0	0	1	0	0	0	1	0	0
Choice	x	5	x	4	x	x	2	x	x	x	6	x	x
Action	00	00	01	01	01	01	11	01	11	10	10	10	11
floor	0	0	1	2	3	4	4	5	5	4	3	2	2

Cycle#	14	15	16	17	18	19	20	21	22	23			
Rst	0	0	0	0	0	0	0	0	0	0			
Valid	0	0	0	0	0	0	1	0	0	0			
Choice	x	x	x	x	x	x	7	x	x	x			
Action	01	01	01	01	11	00	00	01	11	00			
floor	3	4	5	6	6	6	6	7	7	7			