Technical report

Author Mnyar Hees		Task State_machine_door_control	
E-mail address Menyar.hees@gmail.com	Controlled	Version 1	Thread Report_template_C_task_2

State machine

Description of the state machines for the door control with lock

Mnyar Hees

2024-09-23

Summary

This project implements a finite state machine (FSM) that controls an automatic door system with integrated locking functionality. The design uses four states—**OPEN**, **CLOSED**, **LOCKED**, and **ERROR**—and transitions based on synchronized input signals. All push-button inputs are filtered with a two-stage synchronizer to avoid metastability. The system updates three LEDs to indicate the current state and outputs the binary-encoded state for debugging and verification. The complete FSM is fully synchronous, validated through simulation, and structured for reliable hardware implementation.

Content

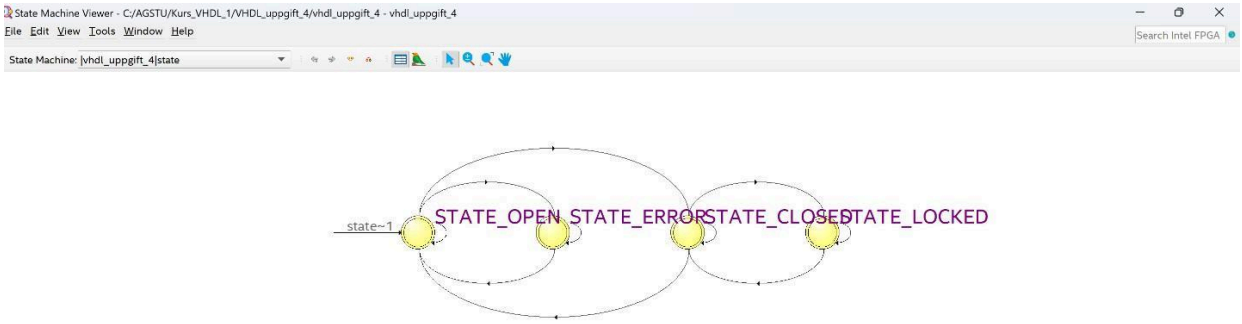
1 Description with state graph

License:

- **STATE_OPEN**: The door is open.
- **STATE_CLOSED**: The door is closed.
- **STATE_LOCKED**: The door is locked.
- **STATE_ERROR**: Error, the door is open but the lock button is pressed.

Transitions:

- From **STATE_OPEN** to **STATE_CLOSED**: When KEY0 is pressed.
- From **STATE_CLOSED** to **STATE_OPEN**: When KEY1 (open) is pressed.
- From **STATE_CLOSED** to **STATE_LOCKED**: When KEY2 (lock) is pressed.
- From **STATE_LOCKED** to **STATE_CLOSED**: When KEY3 (unlock) is pressed.
- From **STATE_OPEN** to **STATE_ERROR**: When KEY2 (lock) is pressed while the door is open.
- From **STATE_ERROR** to **STATE_OPEN**: When reset_n (reset) is activated.



	Source State	Destination State	Condition
1	STATE_CLOSED	STATE_CLOSED	(reg2_KEY2),(reg2_KEY1)
2	STATE_CLOSED	STATE_LOCKED	(!reg2_KEY2),(reg2_KEY1)
3	STATE_CLOSED	STATE_OPEN	(!reg2_KEY1)
4	STATE_ERROR	STATE_ERROR	(reset_n)
5	STATE_ERROR	STATE_OPEN	(!reset_n)
6	STATE_LOCKED	STATE_CLOSED	(!reg2_KEY3)
7	STATE_LOCKED	STATE_LOCKED	(reg2_KEY3)
8	STATE_OPEN	STATE_CLOSED	(!reg2_KEY0)
9	STATE_OPEN	STATE_ERROR	(!reg2_KEY2),(reg2_KEY0)
10	STATE_OPEN	STATE_OPEN	(reg2_KEY2),(reg2_KEY0)

State Table

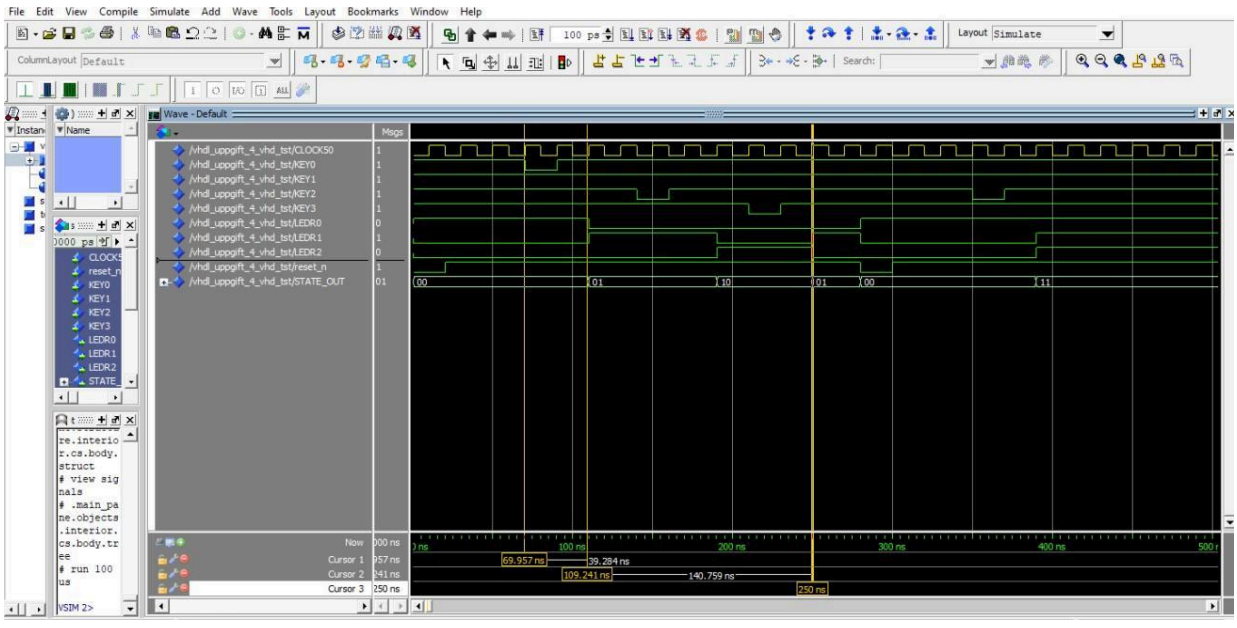
Transitions | Encoding

State Table

	Name	STATE_ERROR	STATE_LOCKED	STATE_CLOSED	STATE_OPEN
1	STATE_OPEN	0	0	0	0
2	STATE_CLOSED	0	0	1	1
3	STATE_LOCKED	0	1	0	1
4	STATE_ERROR	1	0	0	1

Transitions | Encoding

2 Pulse chart of the machine



3 RTL – the state machine

1 **Permit register**:

state: Keeps current state.

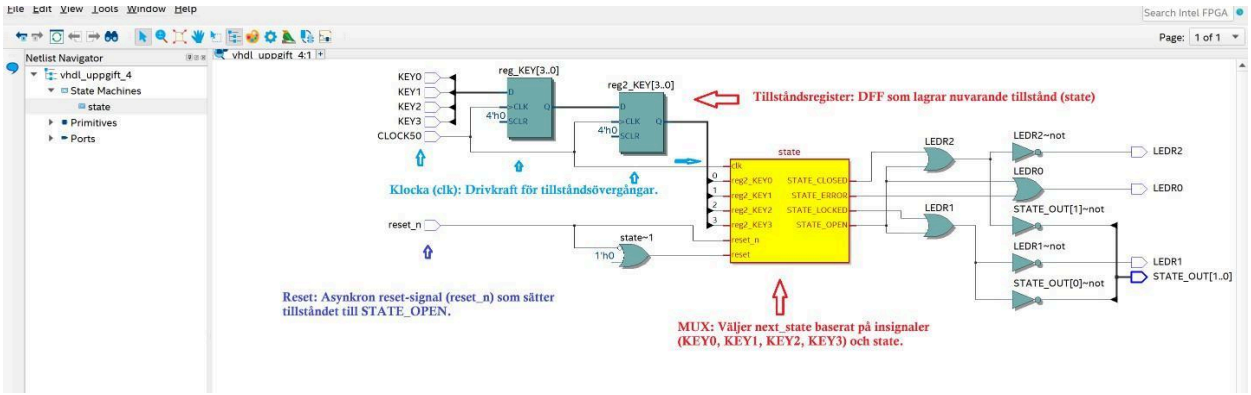
next_state: Holds the next state.

2 **State transition logic**:

A process that takes in current conditions and input signals to set next_state.

3 **Gates**:

Multiplexes (MUX) to select the next state based on the current state and keystrokes. D-flip-flops (DFF) to store the state.



Summary

The state machine for the door control with lock is defined by four states and transitions based on the user's input. The RTL representation uses DFF and MUX to manage state transitions and storage.