

The following code with the lock input answers the quiz question.

```
void combination_lock(
    ap_uint<4> x,
    bool &door_open,
    bool lock) {
#pragma HLS PIPELINE
#pragma HLS INTERFACE ap_none port=x
#pragma HLS INTERFACE ap_none port=door_open
#pragma HLS INTERFACE ap_none port=lock
#pragma HLS INTERFACE ap_ctrl_none port=return

//-----state variables-----
    static state_type state = s0;

    state_type next_state;

//-----temporary output variables-----
    bool door_open_local = 0;

//-----switch case-----
    switch(state) {
    case s0:
        if (x == 2) {
            next_state = s2;
        } else {
            next_state = s0;
        }
        door_open_local = 0;
        break;
    case s2:
        if (x == 3) {
            next_state = s23;
        } else {
            next_state = s0;
        }
        door_open_local = 0;
        break;
    case s23:
        if (x == 4) {
            next_state = s234;
        } else {
            next_state = s0;
        }
        door_open_local = 0;
        break;
    case s234:
        if (x == 6) {
            next_state = s2346;
            door_open_local = 1;
        } else {
```

2 High-Level Synthesis in FPGA, Pat 2 – Sequential Circuits

State Machine-Template: Quiz Solution

www.highlevel-synthesis.com

```
        next_state = s0;
        door_open_local = 0;
    }
    break;
case s2346:
    if (lock == 1) {
        next_state = s0;
        door_open_local = 0;
    } else {
        next_state = s2346;
        door_open_local = 1;
    }

    break;
default:
    break;
}

//-----state and output variable assignments-----
state = next_state;
door_open = door_open_local;
}
```