A.1 General Description of Lift

The lift is a customized structure representing a lift shaft to which are fixed three "Call" buttons and associated "lift coming" indicators, plus a rack-and-pinion gear system and motor whereby a simple platform representing the lift cage can be driven up and down. The position of platform is detected by four more switches- one each for "top", "middle-plus", "middle-minus" and "bottom". The two switches which detect the platform in the middle positon are operated by the rack cage, rather than an isolated point on it – only when both "middle-switches" are closed is the platform properly aligned. The top and bottom position sensing arrangements don't need this refinement because any "up" movement with the "top" switch active is illegal, as is "down" movement with the "bottom" switch active.

The model can be powered using +5V and +12V supplies using a 7 pin DIN connector from an outlet. Simple electronics on a printed circuit board in the model base convert and condition input and outputs, such that each switch provides a logic 'HI' when open and 'LO' when made (but see A.3 below). The platform motor is driven in response to the UP/DOWN "direction" control line when the "/ENABLE" control line is taken 'LO'. An integrated motor drive IC is used to provide appropriate connection of the +12V and 0V supplies to the motor, and to limit the current. Crude speed control is possible by Pulse Width Modulation of the enable control line. Each floor indicator is driven by an active LO-control line.

A.2 Lift Control

The FPGA signal should be connected to the lift model using the ribbon cable adapter supplied, which is connected to header 'J' for reference following as the pin assignment.

Signal	Cable	J pin	FPGA pin
Call0	3	3	138
Call1	5	5	140
Call2	7	7	143
Bottom	9	9	158
Middle-minus	11	11	160
Middle-plus	13	15	162
Top	15	17	164
Direction	16	18	165
Enable	18	20	167

Indicator2	20	22	169
Indicator1	22	24	181
Indicator0	24	26	177
Ground	1,2,17,25,26		

To reset, use the global reset button on the development board.

In the table above, callN and indicatorN refer respectively to the active-LO status call button and active-LO "lift coming" indicator on floor N.

(1) In response to the lift's three "call" buttons, drive the lift to the calling floor and halt there. If a second button is pressed while lift is moving, it is suggested that the controller ignores the second button. If time permits, you can introduce some limited scheduling such that if two buttons are pressed, the lift goes to one floor first and then to the second.
(2) Light a "lift coming" indicator at the calling floor until the lift arrives. (Any response to a "call" when the lift is already at the calling floor is up to you).
(3) Optionally, detect system malfunctions (e.g. Lift or call button stuck) and report them.

All FPGA must be clocked at 20 KHz. This can be obtained from the customized digital test bed.

Warning:

If the lift is in the "middle" position, with both "middle_plus" and the "middle_minus" signals active, the downloading may fail (reason unknown).

Hints and Suggestion:

- How many states must the controller have? The states of the controller should reflect the states of the lift at a floor; rising or falling. Your initial design may contain many more states than are actually needed. If you get this part right, the rest is easy.
- What is the initial state of the lift and initial state of the controller? Therefore, what should a reset do?
- Don't worry about the state machine.
- Debugging the final design may be a problem. The more that you simulate, the surer you can be that your design will work, But if you download the design onto your FPGA and nothing happens, you have very

little information to help you. It is possible to use seven-segment display to show the current state, but this is almost a last resort.

The clock is generated on the FPGA board. A power-on reset is automatically implemented inside the FPGA. An additional, asynchronous, reset should be included.

A3. It's only a model

While there are no safety issues to be addressed in this exercise, there are some real-life aspects. The first and perhaps most obvious is that it takes the lift platform time to move you find it useful to devise testing and development strategies that minimise the impact of this feature.

More importantly, the signals derived from the lift switches are presented with only a minimum of conditioning from mechanical state to logic level. Most real switches "bounce". That is there is a time from first closure (and sometimes first opening) within which the switch state may change again momentarily. The lift switches bounce time is typically 1½ milliseconds (measured when new-they may get worse with age and use). This is negligible relative to the time involved in a lift actually doing anything but is substantial in terms of the number of FPGA clock cycles. This may or may not cause you a problem.

Optional:

Although several protective and limiting circuit features are present in the light interface it is possible that under fault conditions excessive current can be drawn. If that happens, damage is avoided by the digital test-bed (Prepare your own test bed) shutting down its power supply. The fault has to be cleared AND mains power switched off and back on before the power supply will resume normal operation. So if all the digital test-bed LEDs go out, switch it off and seek assistance! The same applies if the lift motor suddenly drives at high speed or if you smell burning.