

### Lab 3 (Week 7 and 8)

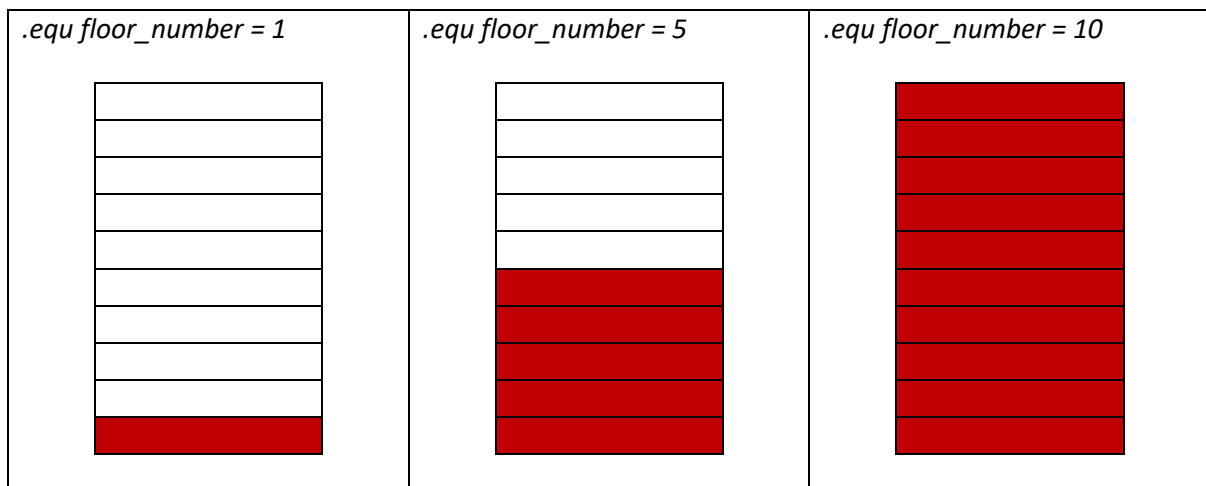
#### Instructions

- Complete each task and demonstrate the working program to your tutor. Tasks should be demonstrated using the board provided.
- **Part A of this lab should be marked during week 7 lab session and the rest during week 8 lab session.**
- The board should be wired as specified in the "[Board test procedure](#)".

#### **Part A – Lift floor display (3 mark)**

Use the LED bar to show the current floor number of the lift. The current floor number can be assumed to be defined by a ".equ" assembler directive as a constant in your source file. Assume there are only 10 floors in the building numbered 1-10. The LEDs should display a filling bar corresponding to the floor number as per the following examples.

Note: Do not worry about the colour of the LEDs, different boards have different coloured LEDs.



#### **Part B – Lift movement (3 marks)**

Use the timer0 and the LED bar to demonstrate the movement of the lift. Assume that the lift takes 2 seconds to travel between floors.

Modify part A such that the floor number is now stored in a register of your choice, which gets updated after 2 seconds. You must use timer0 to generate an interrupt that determines when the floor number and the LEDs are updated. In this exercise, the lift doesn't stop. Instead, it can be assumed to move continuously up and down, i.e., from floor-1 to floor-10, and then from floor-10 to floor-1, and so on. The direction of the lift can be stored in another register as a flag, i.e., 1 or 0.

You should demonstrate the movement of the lift (both up and down) with correct time in between the floors to the tutor for getting full marks.

#### **Part C – Counting push button presses (4 marks)**

Use the left push button on the board to increment an 8-bit counter and the right push button to decrement the counter. The counter must be displayed on the LED bar in binary. Use falling edge external interrupts (INT0 and INT1) to detect the button presses. You can use any suitable general purpose register (r0-r31) as the 8-bit counter.

One button press should increment/decrement the counter by only once. For this, the buttons must be software debounced so that one button press reliably generates only a single trigger. You should implement debouncing by ignoring spurious interrupts. If one press leads to two triggers even after debouncing is implemented, clear the INTn interrupt flag as the last step in the interrupt subroutine - by writing a logical one to its Interrupt Flag bit (INTFn) in the EIFR Register.