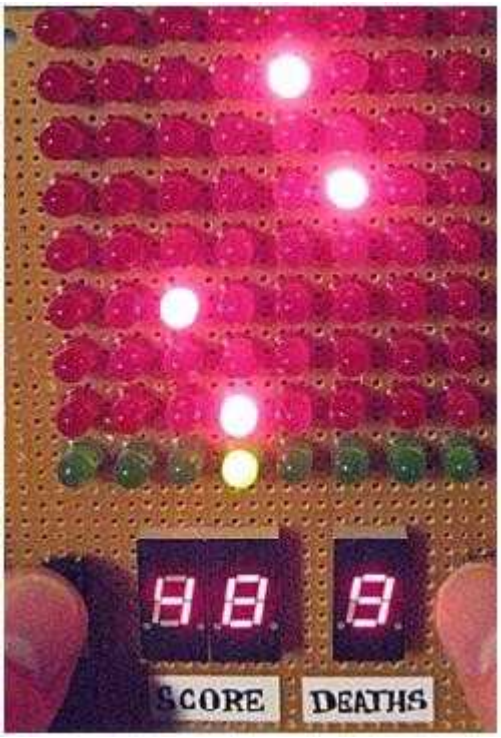
DLD PROJECTS – SPRING 2021

1- Falling Objects Game



Design and implement a game on a grid of 72 LED's (9 rows by 8 columns) in which the user must move a light representing a platform (bucket) from left to right to catch falling objects. The controls are two momentary pushbuttons. If the user successfully keeps the platform under a falling object while it is in the bottom row, the score will increment, shown on the two-digit seven segment display. Otherwise, the death count will increment, displayed on a third seven segment display. There are four different stages in the game, in which the objects fall at different speed, thus increasing the complexity and challenge of the game. The game ends when the user reaches 10 deaths. When this occurs, the game clock is disabled so the objects freeze in place on the screen to indicate that the game is over.

2- Traffic Light Controller

You have to design traffic controller for DLD Square. The configuration of Square is given below.

There are total four roads. Each road has two partitions, one for the traffic coming towards the square (trafficIn) and the other one for traffic going out of the square (TrafficOut). Each partition has three lanes. The cars at right most lane go on the road at right side. The cars in middle lane go straight. The cars in left most lane go on the left side of road. Your task is to implement a system with following requirements:

1- System should show the flow of cars
2- Cars will randomly come on TrafficIn partitions
3- Each lane can have 15 cars at max and you need to display only 4 cars per lane
4- There is one traffic signal per road for trafficIn. TrafficIn will stop if the signal is red and start running when it turns green
5- Four signals (one per road) run clockwise i.e. road 1's signal will be green at start, then road 2's then road 3's, then road 4's and then again road 1's signal will be green
6- Each signal remains green for 16 ticks at max or until there is no car on that road
7- Signal will turn yellow for 4 ticks before turning red

3- Intelligent Elevator System

We have to design Intelligent Elevator System for two lifts and four floors (Basement, Ground, 1st and 2nd floor). The requirements of the system are as follows:

1- In start both lifts will be on ground floor
2- Capacity of both the lifts is 10 people
3- User can call the lift from any floor with two buttons (Up and Down)
4- User's call will be entertained according to following rules:
    a. Lift should detect number of people entering and exiting the lift and it will not move if the number exceeds the capacity
    b. If user at i^{th} floor wants to go Up, the lift at any of (0 to $(i-1)^{th}$) floor and moving in Up Direction will pick him and vice versa
    c. Lift will remain still on a floor if there is no call in the queue
    d. If there was a call on a floor, Lift Door will remain open there for 8 ticks unless it gets door close signal
    e. Upon entering the lift, user will press a button (0-3) to indicate the floor on which he wants to be dropped and door will be closed without waiting for 8 ticks completion
    f. If both lifts are moving in same direction, the one with lesser number of people will follow the call
    g. If both lifts are in same situation then Lift1 will follow the request
    h. If there are maximum people in the lift then no external call will be entertained
5- Movement of the lifts will be displayed according to following rules:
    a. Green Light indicates that lift is standing still on a floor and the door is closed
    b. Red Light indicates that lift has stopped on a floor and its door is open
    c. Blue light indicates that lift is passing by a floor
    d. Two Up and Down Signals will show if a lift is moving up or down
6- Lift should take 8 ticks to cover distance between two consecutive floors

4- 4-bit Processor

Design a 4-bit processor which consists of 4 data registers each of 4 bits and an instruction register (IR) of 7 bits. The processor has four flags ZF, SF, OF and Cout. The first 3 bits of the instruction tells which operation is to be performed, the next 2 bits signifies the first register and the last two bits signifies the second register.

I_6-I_4	I_3-I_2	$I_1 - I_0$
Operation Code	4-bit register operand 1(R1)	4-bit register operand 2 (R2)

The following operations are performed by the processor.

Operation Code	Operation Performed	Description
000	R1 = A	Load the contents of input A in to the register operand 1.
001	R1 = R2	Move the contents of register operand 2 in to register operand 1.
010	R1 = R1 + R2	Add the contents of register operand 1 and register operand 2 and load in register operand 1.
011	R1 = R1 - R2	Subtract the contents of register operand 2 from register operand 1 and load in register operand 1.
100	R1 = R1 * R2	Multiply the contents of register operand 1 and register operand 2 and load in register operand 1 and 2. (As the result is of 8 bits)
101	$R1 = R1/2^i$	Divide the register contents of register operand 1 with 2^i (i is an input) and load the result in register operand 1.
110	$R1 = R1*2^I$	Multiply the register contents of register operand 1 with 2^i (i is an input) and load the result in register operand 1.
111	R1 = R1 + R2	Logical OR the contents of register operand 1 and register operand 2 and load in register operand 1.

Functionality of the flags is described below:

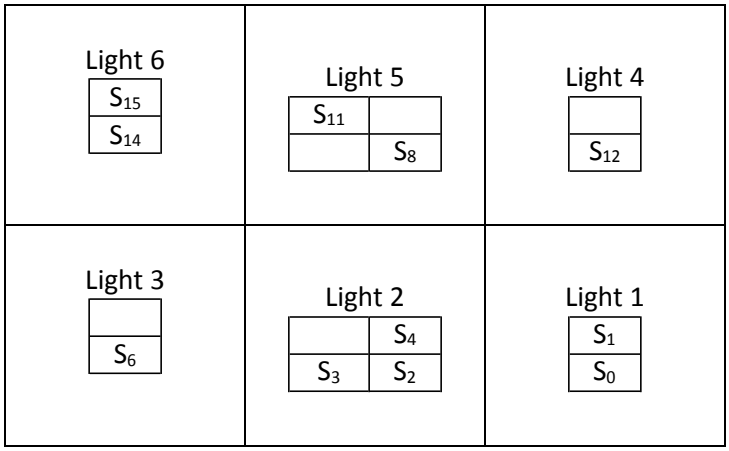
Flag	Operation	Description
ZF	All	Zero Flag: ZF = 0 if result of current ALU operation is zero and 1 otherwise
SF	All	Sign Flag: Reflects the most significant bot of the result of current ALU Operation
OF	Add/Subtract	Overflow Flag: This bit is one if the current operation caused a two's complement overflow – either positive or negative, and will be off otherwise
Cout	Add/Subtract	Carry Flag: This bit is 1 if the addition caused a carry-out from the most significant bit position, so an unsigned overflow

Inputs: Clock Pulse (CP), 7-bits Instruction, A, i

Output: Contents of each register

5- Digital Smart Lock Room

We have to automate six lights and a lock of a room. The room has six enclosures and requirements are as follows:



1- Capacity of room is 16 people
2- Seat plan is shown above and user entering the room will sit on next available seat starting from seat no. 1
3- Only person at seat 1 can exit the room and on his exit all the people will move to previous seat
4- The room should be able to display the number of people inside
5- If all the seats in an enclosure are vacant then its light will be turned off
6- If the room is full its Entry door will be locked and the exit door will be locked if there is no one in the room