LED Space Impact

Abstract:

It is going to be a game where a spaceship has to dodge and finish off space invaders moving towards it at increasing speeds. It is to be made on a set of LED matrices. It draws inspiration from the old mobile game "Space Impact". The space invaders will move horizontally towards the spaceship. The space ship would be given powers to move vertically and fire bullets back at the invaders. The longer a player can survive and the more number of invaders it shoots, the more points he/she gets. Lots of variations and increasing levels of difficulties can be introduced in the game to make it more interesting. We'll use the Krypton board and a Verilog code to implement this idea.

1. Introduction

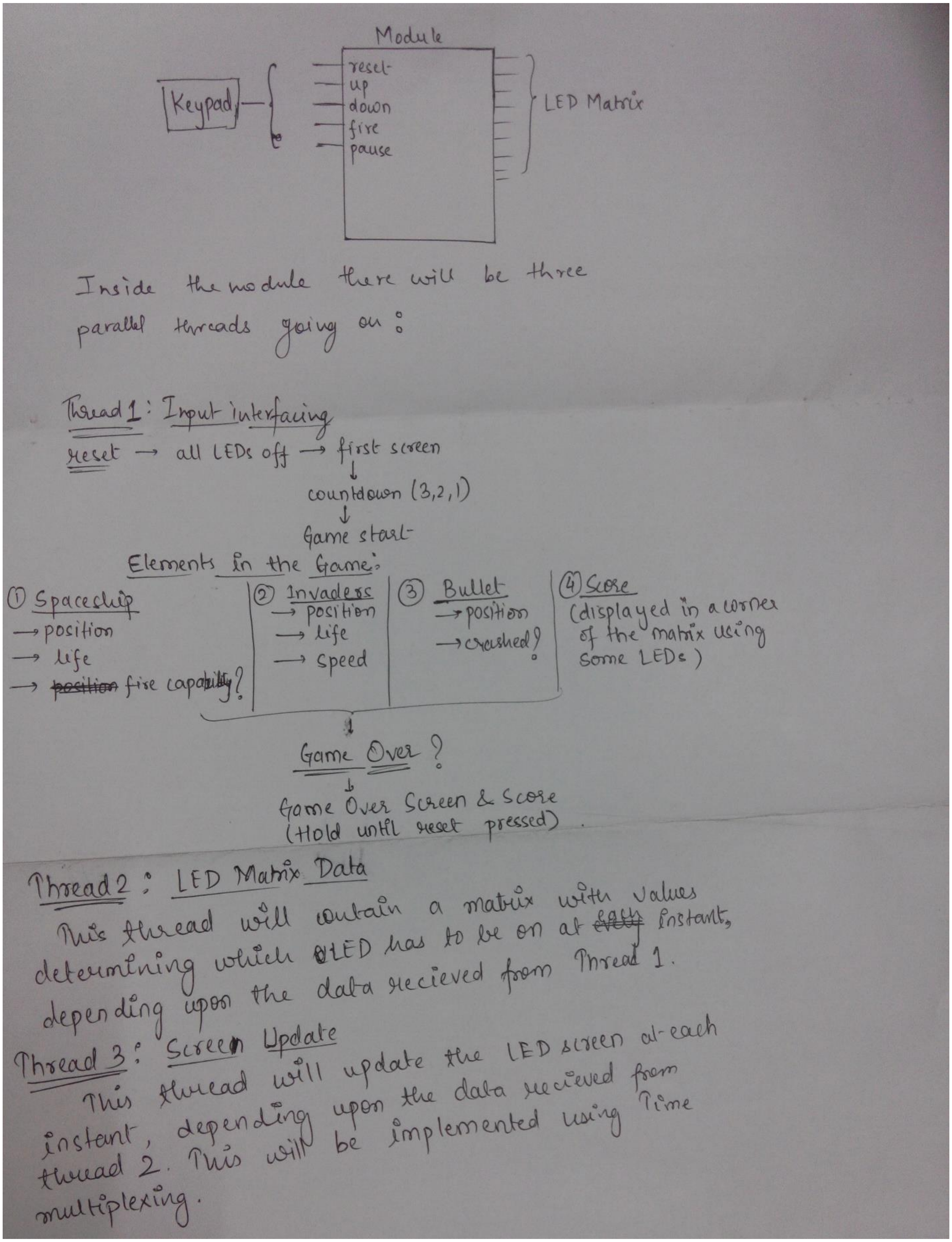
It's the very popular old mobile phone game, where our spaceship, on the left edge of the screen moves past all the invaders charging towards it. We plan to use four to six, 8x8 LED matrices and make specific LEDs glow to represent the Spaceship and invaders and the bullets fired by them, in distinguished shapes. The speed of the invaders moving towards the spaceship will increase with time and they'll appear at discrete vertical levels randomly. They'll move towards the spaceship to crash into it and the player will try to dodge or shoot them to score more points, which we can display using Seven-segment display units. We can make the computer play the game as well. And to increase difficulty we can allow vertical movement of invaders as well and we can also make them fire bullets at the spaceship.

2. Project description

2.1 Project goal:

The goal is to make an enjoyable game. The major challenges we can have is synchronizing the motion of invaders and spaceship over all the LED matrices we use, as we are going to use more than one 8x8 LED matrix. All the LED matrices would be coded appropriately and we'll have to identify when a bullet hits an invader and when an invader crashes into the spaceship and then change scores accordingly. Also if we want an entirely computer controlled game, we have to code the spaceship appropriately so that it can dodge and/or shoot the invaders properly. We'll have to identify the position of each object on the LED module and then write appropriate codes in case they interact.

2.2 Technical Design


Module
Keypad
reset
up
down
fire
pause
LED Matrix

Inside the module there will be three parallel threads going on :

Thread 1 : Input interfacing

reset → all LEDs off → first screen
↓
countdown (3,2,1)
↓
Game start

Elements in the Game :

① Spaceship
→ position
→ life
→ ~~position~~ fire capability?

② Invaders
→ position
→ life
→ speed

③ Bullet
→ position
→ crashed?

④ Score
(displayed in a corner of the matrix using some LEDs)

↓

Game Over ?

↓

Game Over Screen & Score
(Hold until reset pressed).

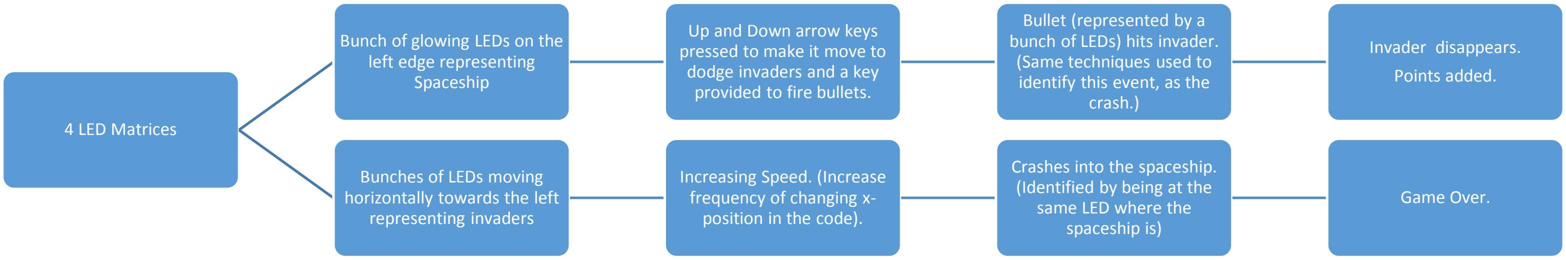
Thread 2 : LED Matrix Data

This thread will contain a matrix with values determining which LED has to be on at ~~every~~ each instant, depending upon the data recieved from Thread 1.

Thread 3 : Screen Update

This thread will update the LED screen at each instant, depending upon the data recieved from thread 2. This will be implemented using Time multiplexing.

2.2.1 System-level overview


4 LED Matrices
Bunch of glowing LEDs on the left edge representing Spaceship
Up and Down arrow keys pressed to make it move to dodge invaders and a key provided to fire bullets.
Bullet (represented by a bunch of LEDs) hits invader. (Same techniques used to identify this event, as the crash.)
Invader disappears.
Points added.
Bunches of LEDs moving horizontally towards the left representing invaders
Increasing Speed. (Increase frequency of changing x-position in the code).
Crashes into the spaceship. (Identified by being at the same LED where the spaceship is)
Game Over.

2.2.2 Performance validation

If we are able to control the spaceship and invaders well and if we are able to fire bullets, identify crashes and increase points well, we'll consider our project successful.

3. Project plan

The coding will take most of our time. Then we'll have to arrange and synchronise the working of all the LED matrices we are going to use. We plan to complete our coding in one third of a time and the finish circuit design and integration and the next third of the time and then test our complete project in the last third.

Milestone 1: Complete Verilog code

Milestone 2: Design Circuit

Milestone 3: Assemble circuit

Milestone 4: Complete testing and Debugging

GANTT CHART

Task Name	**Due Date**	**Will be done by**	**Remark**
1. **Writing Code**			
1.1 Thread 1	10 March	Both	
1.2 Thread 2	6 March	Ashish	
1.3 Thread 3	8 March	Utkarsh	
1.4 Final Compilation	17 March	Ashish	
2. Circuit Design	18 March	Utkarsh	
3. Assembling Circuit			
3.1 Procuring Hardware	19 March	Both	
3.2 Keypad Interfacing	19 March	Ashish	
3.3 LED Matrix Interfacing	20 March	Both	
3.4 Final Circuit Integration	24 March	Utkarsh	
4. Testing and Debugging			
4.1 Code testing, debugging and simulation	27 March	Ashish	
4.2 Circuit testing and debugging	30 March	Utkarsh	

4. Project Implementation:

I don’t think we’ll need any major instruments except for the ones needed to hook up the LED matrices with the Krypton boards and the controlling keys.
Hardware we’ll need:

1. 4 to 6 LED Matrices
2. Krypton Board
3. Connectors for LED Matrices