

# TETRIS GAME

## Project Report

We have successfully solved the problem statement and all the functions that were mentioned in it, as you can see from the video attached along with this document and the design file itself.

The circuit seems to take a lot of time in some steps but every step just takes a single clock cycle to execute(Which is set at 1 Hz). It is the proteus software which lags due to the complexity of the circuit. Please look at the clock(animating clock of proteus) at the bottom left of the window while watching the video.

The Key features of our digital design are-

- It can perform all the function mentioned in the PS. Namely- left transition, right transition and skipping of a row.
- It takes care of the boundary conditions and do not respond to the command of moving left, right or skipping a row wherever these actions are not possible.
- Our design can also recognise when the row has become all 1s and destroys that row and move all the rows above it down by one step before accepting any new input.
- It accepts only one input at a time from the ROM.
- All the actions are performed in a single clock cycle. This includes the transition of the blocks, left/right/skip motion or destroying of row and shifting the rows above it.

We have considered all the 80 blocks individually and each has a state which indicated that it is frozen or not. At the start all the blocks are unfrozen and the state of the blocks are updated everytime a new block

settles at any place in the grid.

It was hard to keep the track of incoming bits individually. So instead of moving just the incoming 1s in the input, we move the whole row to the next row.

Only the unfrozen blocks of the rows are moved to the next row and the frozen blocks do not accept any input. So it seems like that only the input bits are moving.

We decide when to settle the block by taking bitwise-AND of all the adjacent rows. As soon as any unfrozen block with value 1(i.e. input), comes above the frozen block with value 1, the circuit is reset and the input is frozen and new input is accepted from the ROM.

Also as soon as any row becomes all 1s, that row is destroyed and all the rows above it are shifted downwards. We have implemented it by keeping the record whether a row is all 1s or not. If a row is found to be all 1s, then all the rows above it are unfrozen and are allowed to move by one step.

Our primary concern was to solve the problem statement and implement all the possible functions but at the background we have also tried to keep the cost under control as much as possible. But we did not compromise performance for the cost.

There are a lot of small small steps that we took to make our digital design perfect and perform all the functions given in the PS. It is not possible to go into the details of all of them. You can contact us if you have any query or doubt about our circuit.

Regards

Shatakshi Gupta

2019A3PS0397H