CS223 – Cellular Automata Project Report

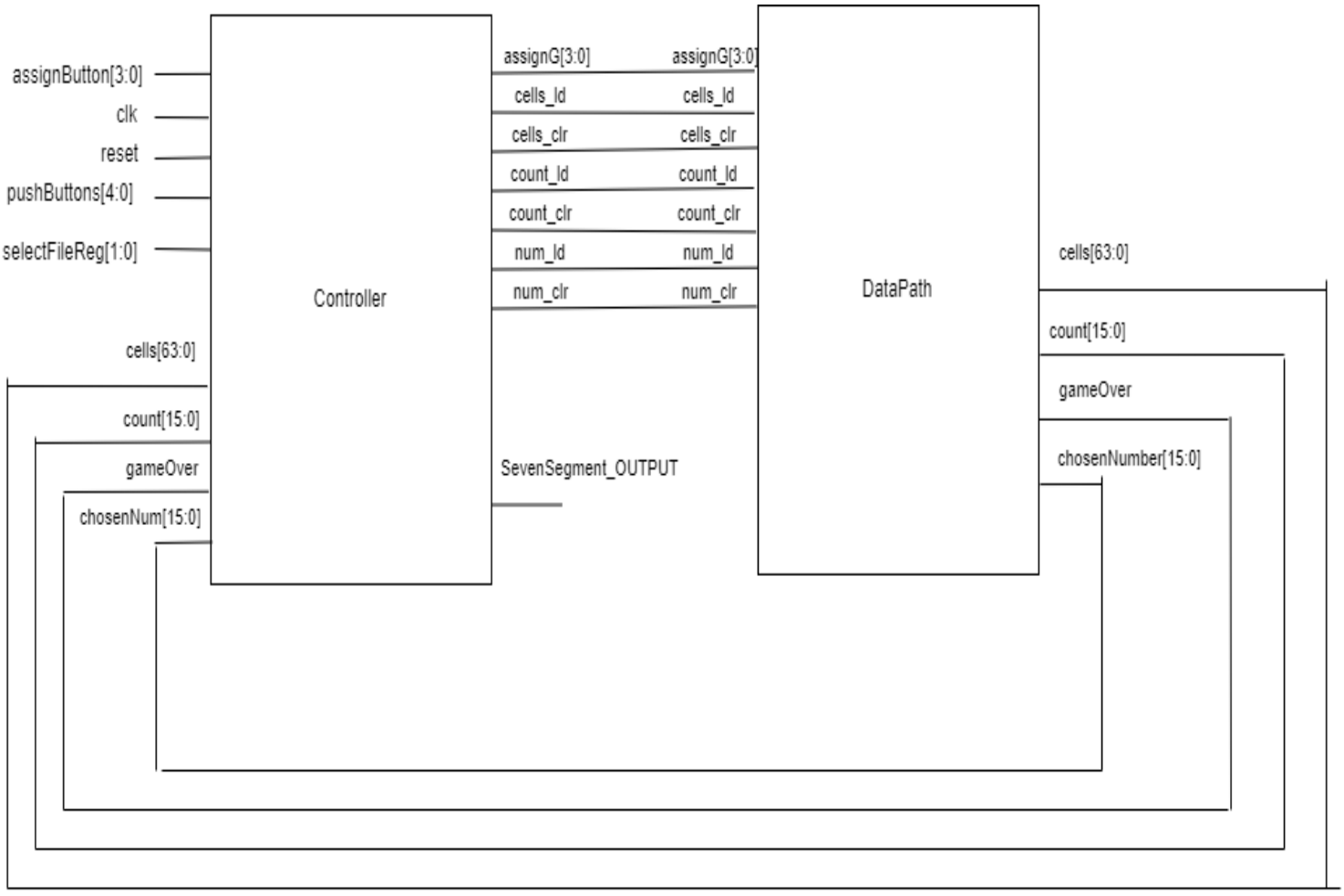
Section: 01
Student name & surname: Uğur Erdem Seyfi
ID : 21801744
Date of Submission : 30.12.2019

Brief Introduction

In this project, a cellular automata game (which its rules are derived from studen ID number) has been built. The game is built by using High Level State Machines.

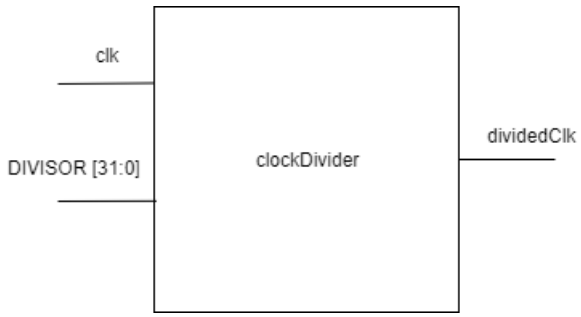
Block Diagram

This is the HLSM block diagram, in other words, top view block diagram of the program:



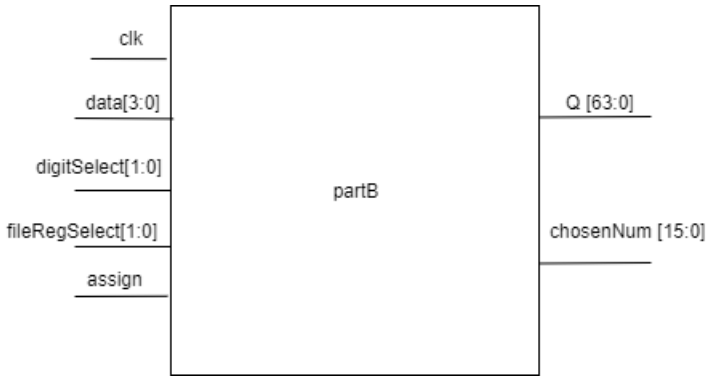
Both controller block and data_path block uses other blocks such as: clockDivider, SevSeg_4Digit, converter, partB, cellsProcess, cellsGroupProcess. Now we proceed to explain what each of these blocks do:

clockDivider block



This module takes a clk and a [31:0] bus to denote the which number to divide by.

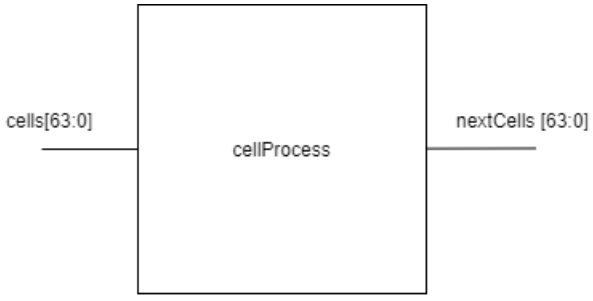
partB block



this block takes clk as an input for the registers in the block. Data[3:0] switches denotes the digit number, digitSelect[1:0] switches for chosing the digit, fileRegSelect[1:0] is for chosing the entire 4digit numbers, assign for the assigning the current data to the current digit value.

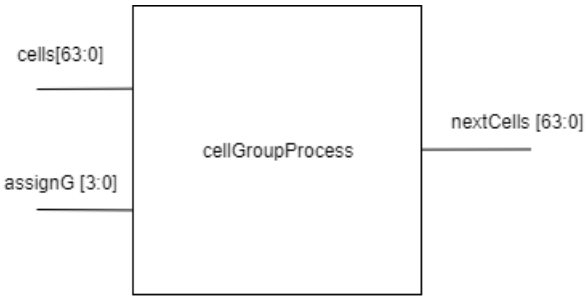
Output Q[63:0] is basically denotes the whole numbers that are saved.
Output chosenNum [15:0] denotes the number choosen by fileRegSelect[1:0] switches.

cellsProcess block



this block is a combinational logic block that takes [63:0] cells input and by using the rules specified by ID number, outputs the next state of all cells.

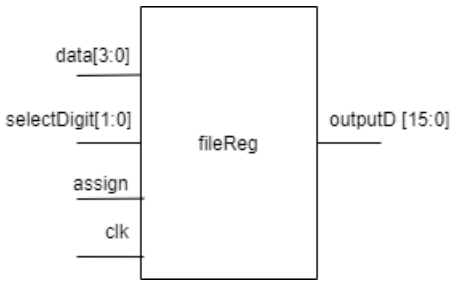
cellsGroupProcess block



this block, again takes cells input but it also takes another bus input assignG[3:0] which is used to decide which group of cells should be updated. This block only updates the cells if they are chosen.

FileReg block

This block takes clk, _assign, data[3:0], selectDigit[1:0] as inputs, this block is used for storing 4 digit (hexademical) numbers. Its output, outputD [15:0] is the stored 4 digit hexadecimal number.



SevSeg_4Digit block

This block is given to us, but I made a small change in the block. I take one more input called enable, the SevSeg_4 displays nothing when enable is 0. This port is used in order to make blinking process easier.

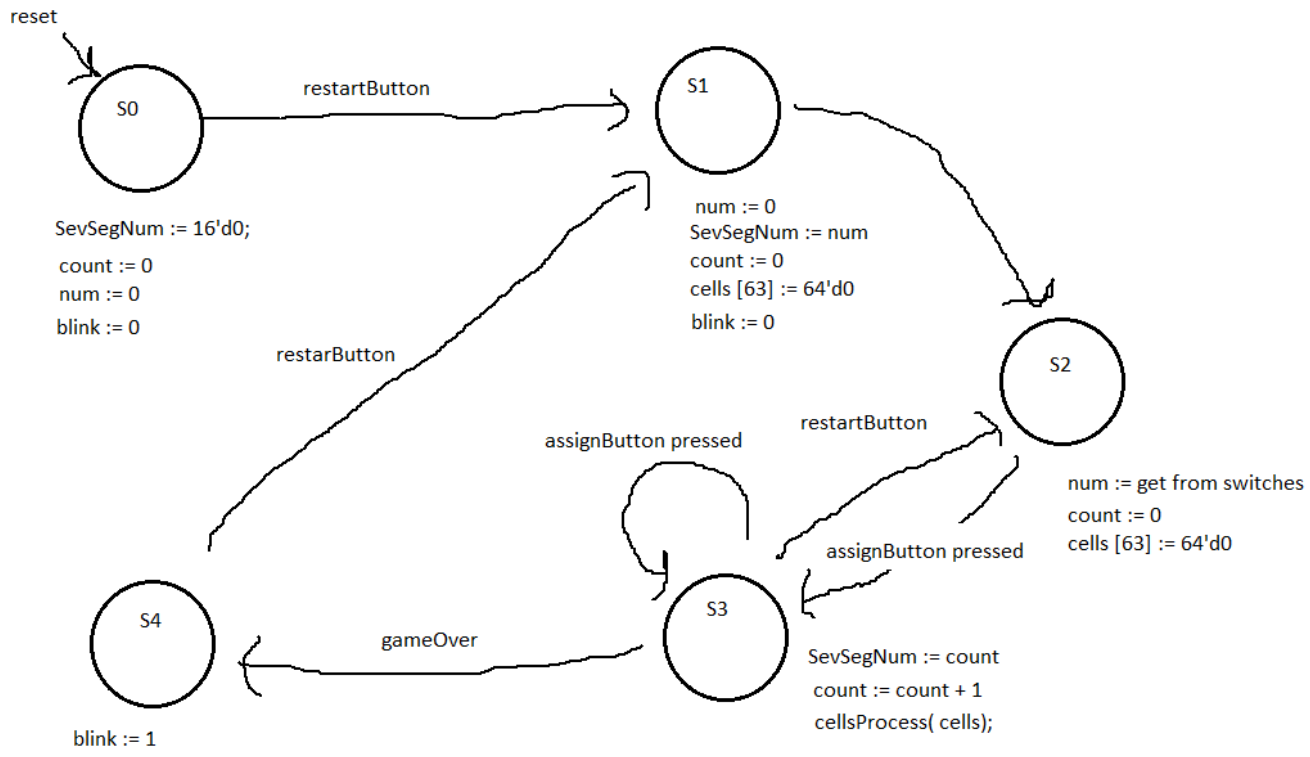
converter block

This block is given to us. This block takes [7:0] bus of [7:0] busses, then outputs the values that is needed to display this [7:0][7:0] bus on a grid.

Detailed Explanation of the Work

1. HLSM design

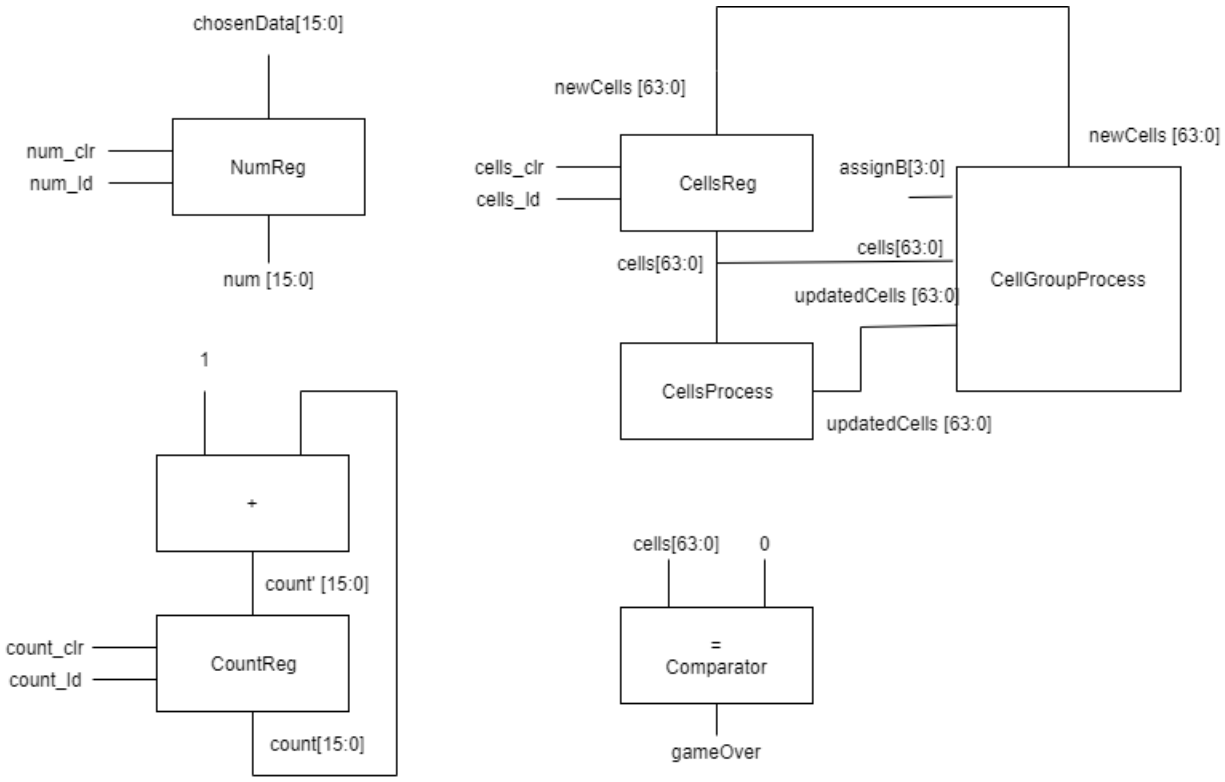
This is the simple HLSM diagram for the program. State transitions and outputs are handled by controller block, other processes are done in data_path block.



2. Explanations of block implementations

Explanations of the block input/outputs are already done in the block design part. Now we explain how the inner circuits of some of the important blocks are implemented.

data_path block inner circuit explanation

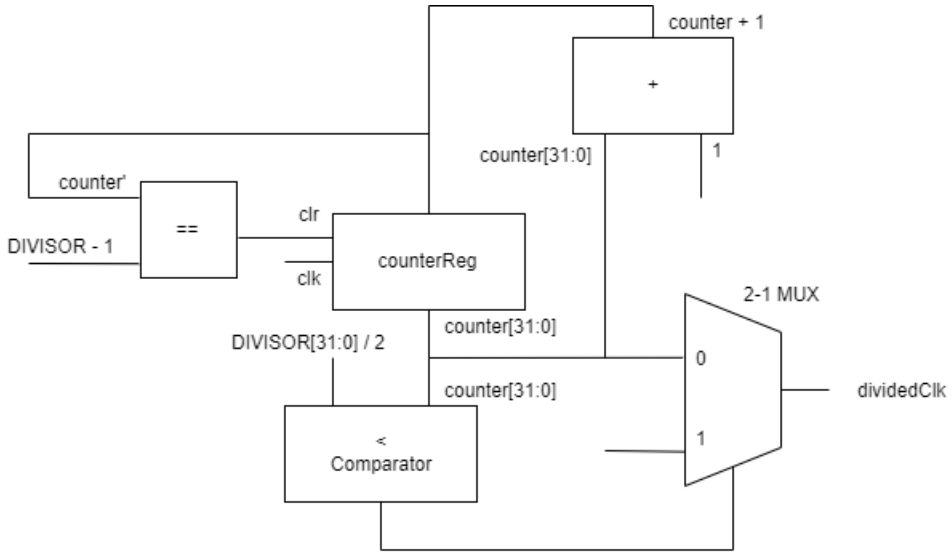


here CountReg, NumReg and CellsReg are registers with load and clr. In all registers except CellsReg, clr returns the output 0, in CellsReg clr returns the output of numMemory[63:0].

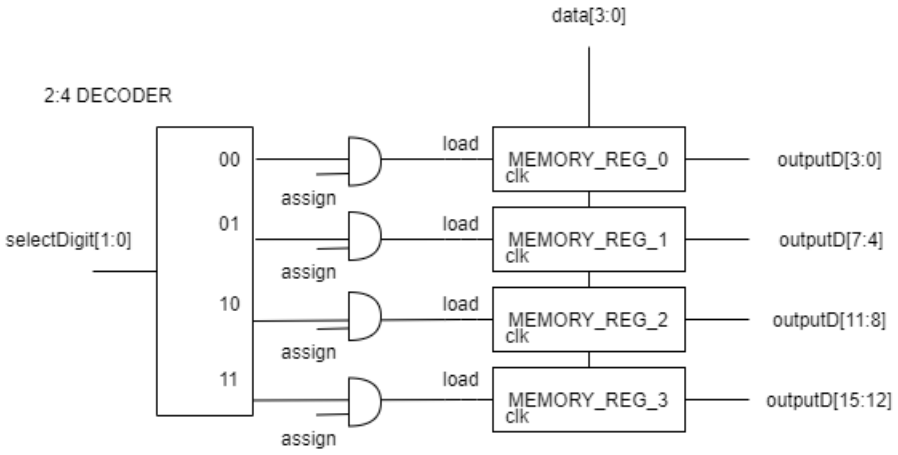
controller block explanation

Implementation of this block is nearly the same as implementing an FSM. We have some outputs and states and we implement the controller block in the same way we do for FSM by implementing the next state / output truth table.

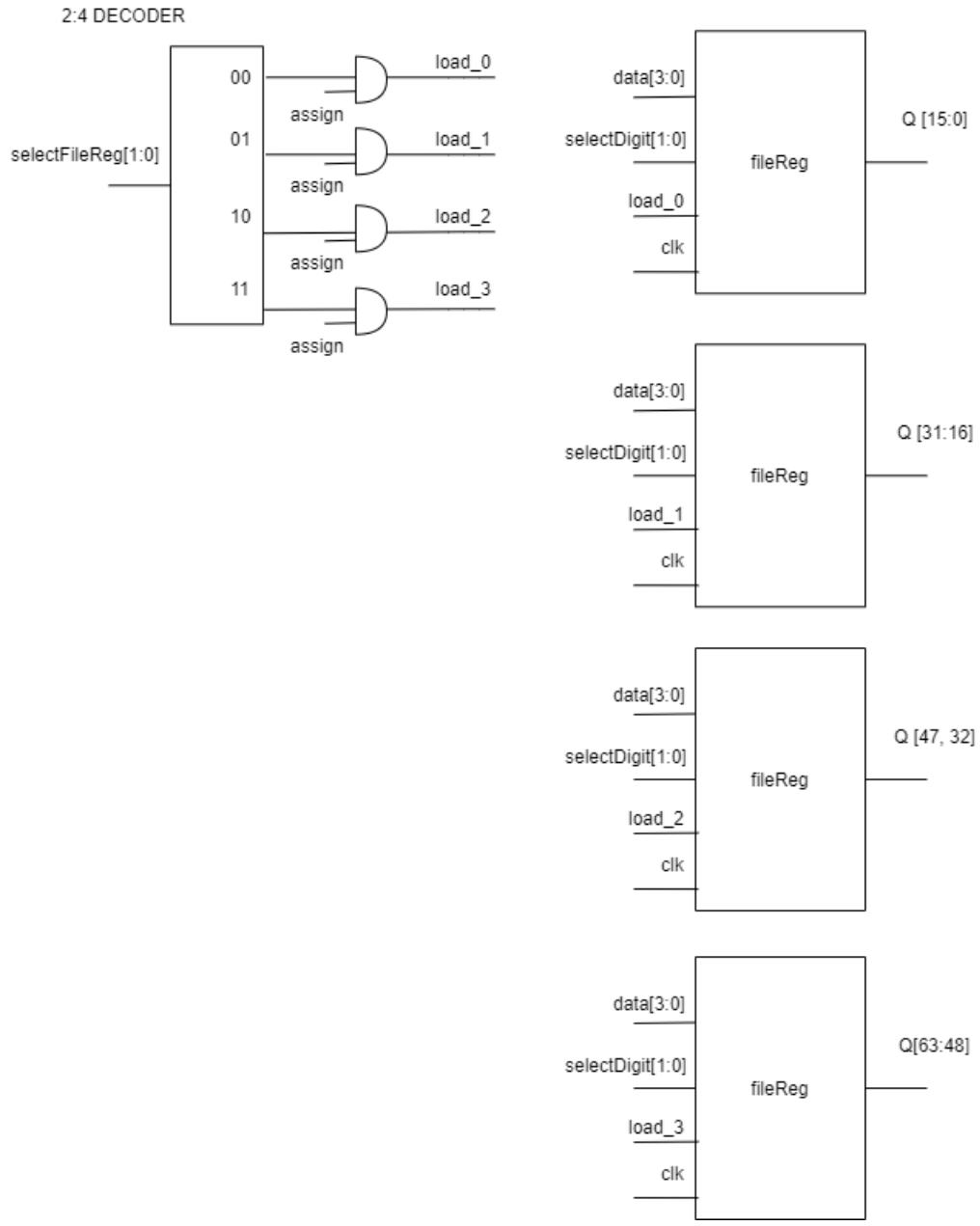
clockDivider block explanation



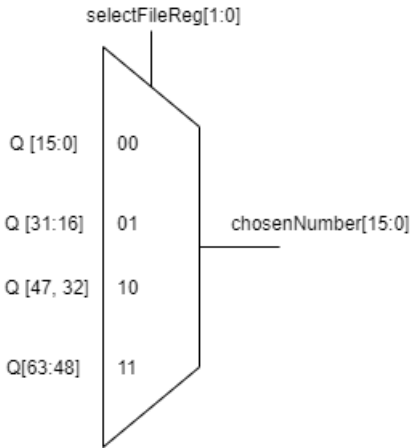
fileReg block explanation



partB block explanation



and the chosenNum[15:0] is chosen by the following multiplexer



cellsProcess and cellGroupProcess blocks explanation

Both of these blocks includes a combinational logic, however since there are 64 different cells that are determined by 2 level combinational logic in each of the blocks, it is hard to show the implementation via schematic.

CONCLUSION

Now we have a HLSM cellular automata game that determines the initial situation of cells by taking 4, 4-digit hexadecimal based numbers, then counts and proceeds as the user pushes button, and goes to a gameOver state when all cells are “dead”.