

Verilog Sudoku Solver

Magson Gao, Shreeyam Kacker

Abstract

The aim of this project is to design and implement a Sudoku solver using digital logic written in Verilog HDL. In order to implement this solver it is necessary that the digital logic is aware of the rules of Sudoku, several key techniques for harder Sudoku and when it is necessary, how to guess values for cells in the Sudoku grid. The techniques that will be implemented include, but are not limited to, the candidate line, naked group and hidden group methods. Whilst these methods are sufficient for solving most novice to intermediate level Sudoku's, the most difficult Sudoku require backtracking. The previous guesses will be stored on a stack based architecture to allow for previous states of the grid to be restored. The purpose of implementing the aforementioned techniques is to significantly reduce the search space required for guessing, allowing the solution to be found in fewer clock cycles and requiring less memory to store previous guesses for the typical Sudoku. The FPGA that will be used is that on the Digilent Nexys 4 development board.

The system will be evaluated by a testbench containing a set of multiple randomly generated Sudoku for each difficulty level. Each Sudoku will be generated using the QQWing Sudoku generator (<https://qqwing.com/generate.html>) as it provides not only the Sudoku itself but a list of the required algorithms used to solve the Sudoku and the number of required guesses. The testbench will also include the "World's Hardest Sudoku" (Arto Inkala, 2010) which is one of the approximately, 50,000 minimum hint Sudoku each of which contain only 17 of the 81 cells pre-filled in (McGuire, G. et al, 2012, <https://arxiv.org/abs/1201.0749>).

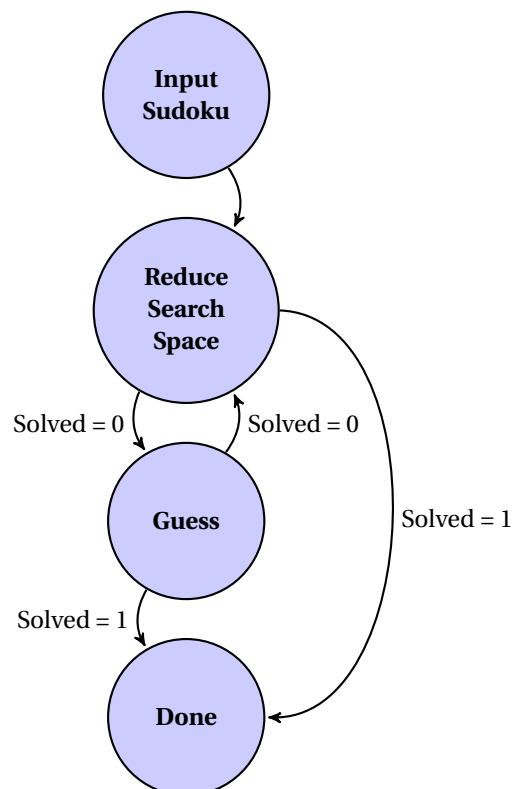


Figure 1: The high level functional diagram for the Sudoku solver.