FPGA LAB ASSIGNMENT 1

TADIPATRI UDAY KIRAN REDDY
EE19BTECH11038

January 16, 2022

Problem

Obtain the minimal form for the following Boolean expression using Karnaugh's Map.

$$H(P, Q, R, S) = \sum(0, 1, 2, 3, 5, 7, 8, 9, 10, 14, 15)$$

Solution

After simplification of the above truth table with Karnaugh's map shown in Figure 1, we get the following boolean expression

$$H = Q'S' + Q'R' + P'S + PQR$$

We see that min terms are reduced from 11 to 4!. And corresponding NAND implementation is shown in Figure 2.

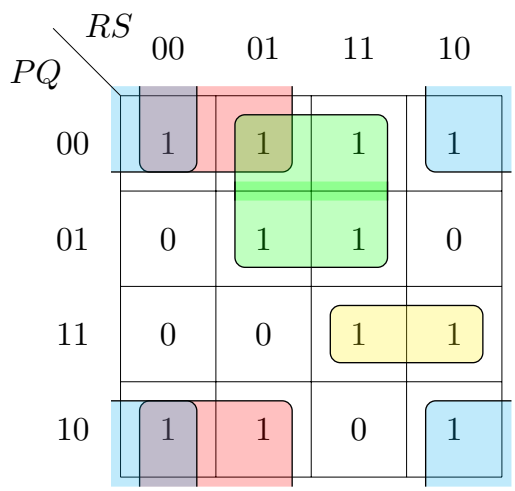


Figure 1: K-Map for H.

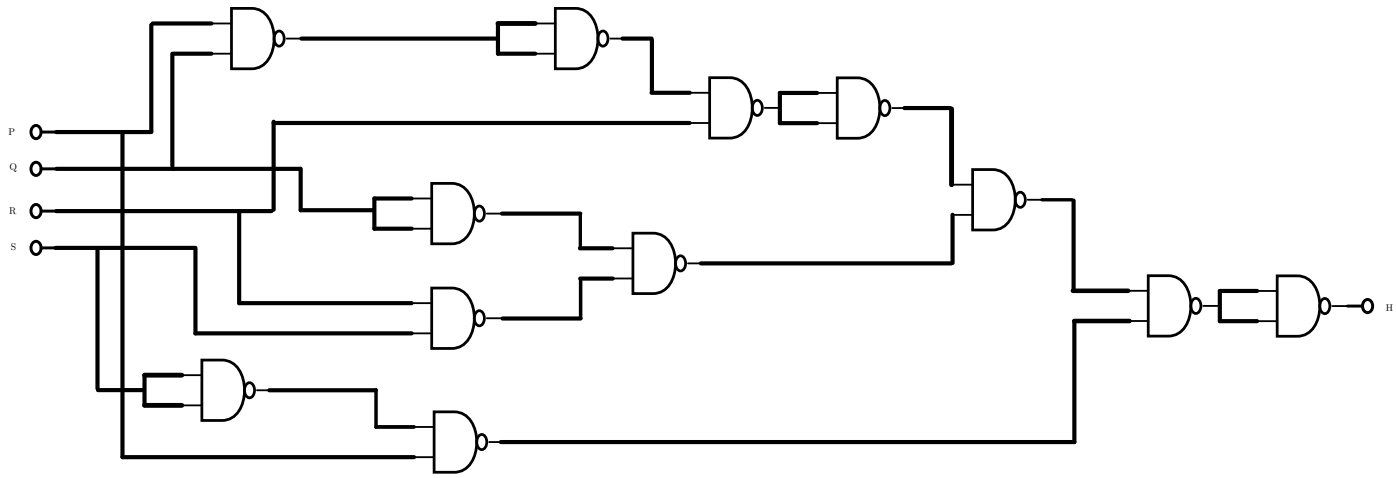
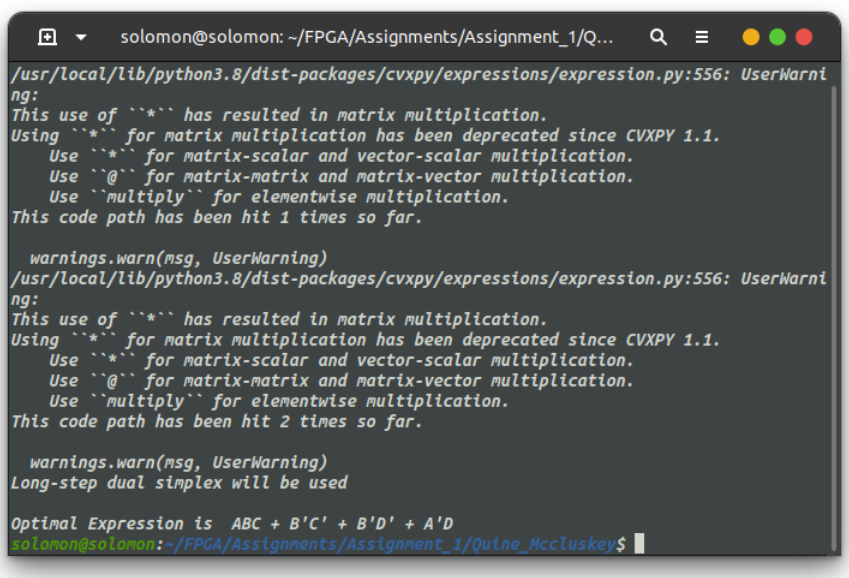


Figure 2: NAND implementation of H.

Optimality verfication

To verify the optimality of above result, The given min terms are given to *Quine-McCluskey* algorithm implemented here. This was implemented using *cvxpy*.



```
solomon@solomon: ~/FPGA/Assignments/Assignment_1/Q...
/usr/local/lib/python3.8/dist-packages/cvxpy/expressions/expression.py:556: UserWarni
ng:
This use of ``*`` has resulted in matrix multiplication.
Using ``*`` for matrix multiplication has been deprecated since CVXPY 1.1.
    Use ``*`` for matrix-scalar and vector-scalar multiplication.
    Use ``@`` for matrix-matrix and matrix-vector multiplication.
    Use ``multiply`` for elementwise multiplication.
This code path has been hit 1 times so far.

  warnings.warn(msg, UserWarning)
/usr/local/lib/python3.8/dist-packages/cvxpy/expressions/expression.py:556: UserWarni
ng:
This use of ``*`` has resulted in matrix multiplication.
Using ``*`` for matrix multiplication has been deprecated since CVXPY 1.1.
    Use ``*`` for matrix-scalar and vector-scalar multiplication.
    Use ``@`` for matrix-matrix and matrix-vector multiplication.
    Use ``multiply`` for elementwise multiplication.
This code path has been hit 2 times so far.

  warnings.warn(msg, UserWarning)
Long-step dual simplex will be used

Optimal Expression is  ABC + B'C' + B'D' + A'D
solomon@solomon:~/FPGA/Assignments/Assignment_1/Quine_Mccluskey$
```

NOTE:- Here A, B, C, and D corresponds to P, Q, R, and S respectively.

Boolean expression verification

A testbench was created using NAND logic to verify the correctness of the obtained boolean expression.