Computer Assignment 2
ECE2300 Digital Logic Design, Section 2
Instructor: Adrian Gonzalez

Kyler Martinez
October 4th, 2020

Objective

The objective of the second computer assignment is for students to become familiar to MATLAB and see how MATLAB can be used to help analyze circuits using logic elements. Another objective of the computer assignment is to gain practice using four variable K-maps to minimize expressions and to use PSPICE to simulate logic circuits and compare the simulated results with that of the original truth table.

Results

Output of MATLAB Code with f(w,x,y,z) = x'y+x'zy'+wy'x+wyz'+x'y'z'+xz'

```
 w x y z | f
-------------
 0 0 0 0 | 1
 0 0 0 1 | 1
 0 0 1 0 | 1
 0 0 1 1 | 1
 0 1 0 0 | 1
 0 1 0 1 | 0
 0 1 1 0 | 1
 0 1 1 1 | 0
 1 0 0 0 | 1
 1 0 0 1 | 1
 1 0 1 0 | 1
 1 0 1 1 | 1
 1 1 0 0 | 1
 1 1 0 1 | 1
 1 1 1 0 | 1
 1 1 1 1 | 0
f(w,x,y,z) = SUM m(0,1,2,3,4,6,8,9,10,11,12,13,14)
```

Figure 1: Output of MATLAB code found in Appendix

K-Map Simplification

wx \ yz	00	01	11	10
00	1	1	1	1
01	1	0	0	1
11	1	1	0	1
10	1	1	1	1

Table 1: Filled K-Map without grouping

wx \ yz	00	01	11	10
00	1	1	1	1
01	1	0	0	1
11	1	1	0	1
10	1	1	1	1

Table 2: Filled K-Map with grouping

From the K-Map, f(w,x,y,z) = z' + x' + wy'

PSPICE Simulation Results

Note: The full graph is displayed only once to reduce the amount of space used. Also, the time was extended to 18s to ensure there was extra room on the graph so all values can be seen.

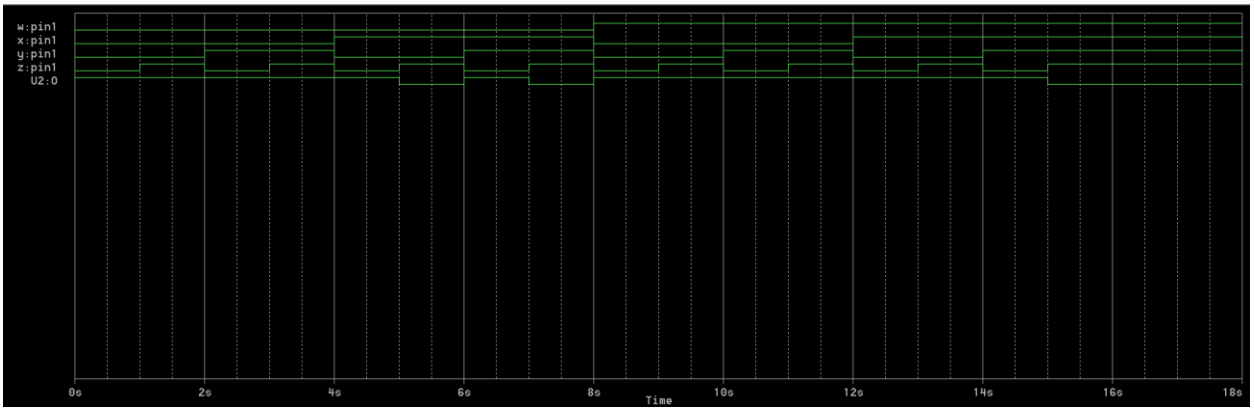

Figure 2: Graph of output f after running the simulation in PSPICE from 0s to 18s

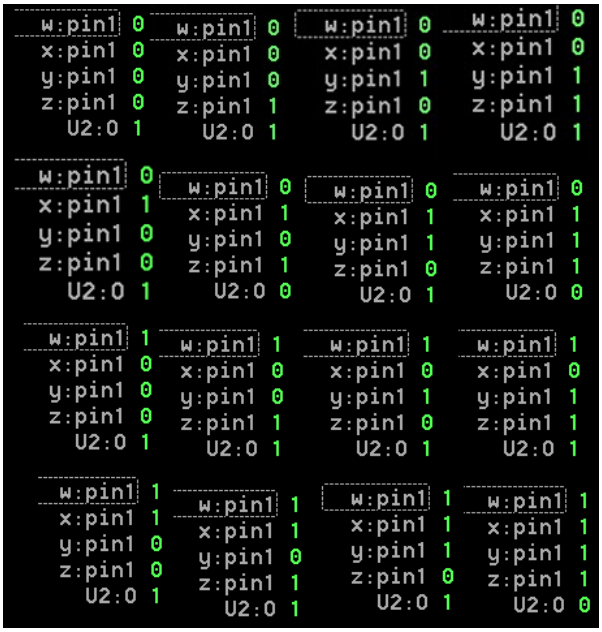

Figure 3: Image compiling all results

w	x	y	z	f(w,x,y,z)
0	0	0	0	1
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0

w	x	y	z	f(w,x,y,z)
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	0

Table 3: Truth table made of the results from simplified expression

Analysis

We can see that the truth table produced by the MATLAB code is the same as the one produced by K-MAP simplified function. This showcases that K-Maps is a tool that at the bare minimum succeeds in producing an alternative function that will produce the same truth table. This is seen in the simplified expression of f(w,x,y,z) which contains four literals, as opposed to six-teen. This also reduces the number of necessary devices that would have been needed to implement the circuit in PSPICE, and with physical devices. The original expression would have needed, three NOT gates, two, two-input AND gates, four three-input AND gates, and one six-input OR gate. However, a six-input OR gate would need to be divided into multiple OR gates most likely. The simplified expression would need three NOT gates, one two-input AND gates, and one three-input OR gate. The original expression would need fifteen gates at the minimum while the simplified expression would need five gates at the bare minimum. This shows how the needed materials have greatly decreased.

Conclusion

Overall, it is apparent that K-Map simplification is an effective tool in simplifying logic functions. This can result in a function that is easier to analyze at face value and implement due to decreasing the cost of implementation. This is the direct result of reducing the amount of logic gates needed. MATLAB and PSPICE are good tools to help in analyzing logic functions by being able to find the truth table of a function and then use PSPICE to implement a function and in turn be able to discern if the function is producing desirable effects.

Appendix

MATLAB Code Used:

```
clc; clear; close all;
n=4; %Number of inputs
inputs = zeros(2^n,4); %pre-allocate 2-d array
for i=0:2^n-1
    binStr = dec2bin(i,4); %Map each combination to binary
    for j=1:n
        inputs(i+1,j) = str2double(binStr(j)); %Fill each row
    end
end
w=inputs(:,1); x=inputs(:,2); y=inputs(:,3); z=inputs(:,4); %Extract each column
f=zeros(2^n,1); %Pre-allocate function
for i=1:2^n %Evaluate f for each combinational input
    f(i) = ~x(i)&y(i) | ~x(i)&z(i)&~y(i) | w(i)&~y(i)&x(i) | ...
        w(i)&y(i)&~z(i) | ~x(i)&~y(i)&~z(i) | x(i)&~z(i);
    % | means OR
    % & means AND
    % ~ means NOT
end

som='f(w,x,y,z) = SUM m(';
%Print table:
fprintf(' w x y z | f \n');
fprintf('-------------\n');
for i=1:2^n
    fprintf(' %d %d %d %d | %d\n', w(i), x(i), y(i), z(i), f(i));
    if(f(i))
        som=strcat(som,num2str(i-1),',');
    end
end
som=strcat(som(1:end-1),')');
fprintf('%s \n',som);
```

Figure 4: MATLAB code used to find the truth table of f(w,x,y,z)

PSPICE Circuit Schematic:

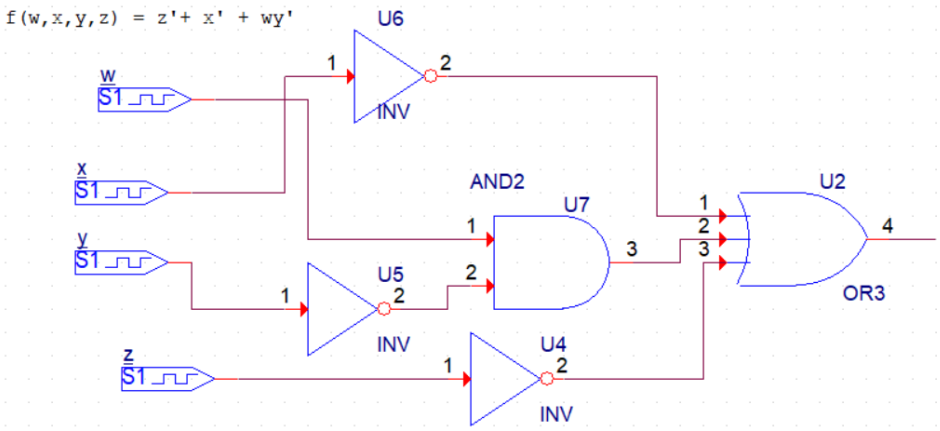

Figure 5: Implementation of f in PSPICE, based on K-map simplification