**컴퓨터공학 설계 및 실험Ⅱ**

8주차 결과보고서

서강대학교 공학부 컴퓨터공학 전공

20171646 박태윤

**1. 실험 목적**

우선 7-Segment Display의 개념을 이해하고, 진리표를 작성한 뒤 이를 토대로 카르노 맵을 만들어 식을 간소화시킨다. 이후 Verilog로 7-Segment Display를 구현하여 Simulation을 통해 결과를 확인한다. 마지막으로 FPGA를 통해 구현한 회로가 제대로 동작하는지 확인한다.

**2. 7-Segment Display의 결과 및 Simulation 과정에 대해서 설명하시오.(Truth table 작성 및 k-map 포함,0~F,DP)**

- 진리표

7-Segment Display의 진리표는 다음과 같다.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 16  진수 | 입력 | | | | 출력 | | | | | | | |
| w | x | y | z | A | B | C | D | E | F | G | Dp |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 2 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 3 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
| 4 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| 5 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 | 1 | 1 |
| 6 | 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 |
| 8 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 9 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |
| A | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
| B | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| C | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 |
| D | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| E | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| F | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |



진리표는 위의 그림을 따라 작성하였다.

- 카르노 맵

(A)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **wx / yz** | **00** | **01** | **11** | **10** |
| **00** | 1 | 0 | 1 | 1 |
| **01** | 0 | 1 | 1 | 1 |
| **11** | 1 | 0 | 1 | 1 |
| **10** | 1 | 1 | 0 | 1 |

A = x’z’ + w’y + xy + wy’z’ + wx’y’ + w’xz

(B)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **wx / yz** | **00** | **01** | **11** | **10** |
| **00** | 1 | 1 | 1 | 1 |
| **01** | 1 | 0 | 1 | 0 |
| **11** | 0 | 1 | 0 | 0 |
| **10** | 1 | 1 | 0 | 1 |

B = x’z’ + w’x’ + w’y’z’ + w’yz + wy’z

(C)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **wx / yz** | **00** | **01** | **11** | **10** |
| **00** | 1 | 1 | 1 | 0 |
| **01** | 1 | 1 | 1 | 1 |
| **11** | 0 | 1 | 0 | 0 |
| **10** | 1 | 1 | 1 | 1 |

C = w’y’ + w’z + y’z + w’x + wx’

(D)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **wx / yz** | **00** | **01** | **11** | **10** |
| **00** | 1 | 0 | 1 | 1 |
| **01** | 0 | 1 | 0 | 1 |
| **11** | 1 | 1 | 0 | 1 |
| **10** | 1 | 0 | 1 | 0 |

D = w’x’z’ + wy’z’ + x’yz + xy’z + xyz’

(E)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **wx / yz** | **00** | **01** | **11** | **10** |
| **00** | 1 | 0 | 0 | 1 |
| **01** | 0 | 0 | 0 | 1 |
| **11** | 1 | 1 | 1 | 1 |
| **10** | 1 | 0 | 1 | 1 |

E = x’z’ + yz’ + wx + wy

(F)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **wx / yz** | **00** | **01** | **11** | **10** |
| **00** | 1 | 0 | 0 | 0 |
| **01** | 1 | 1 | 0 | 1 |
| **11** | 1 | 0 | 1 | 1 |
| **10** | 1 | 1 | 1 | 1 |

F = y’z’ + wx’ + wy + w’xy’ + xz’

(G)

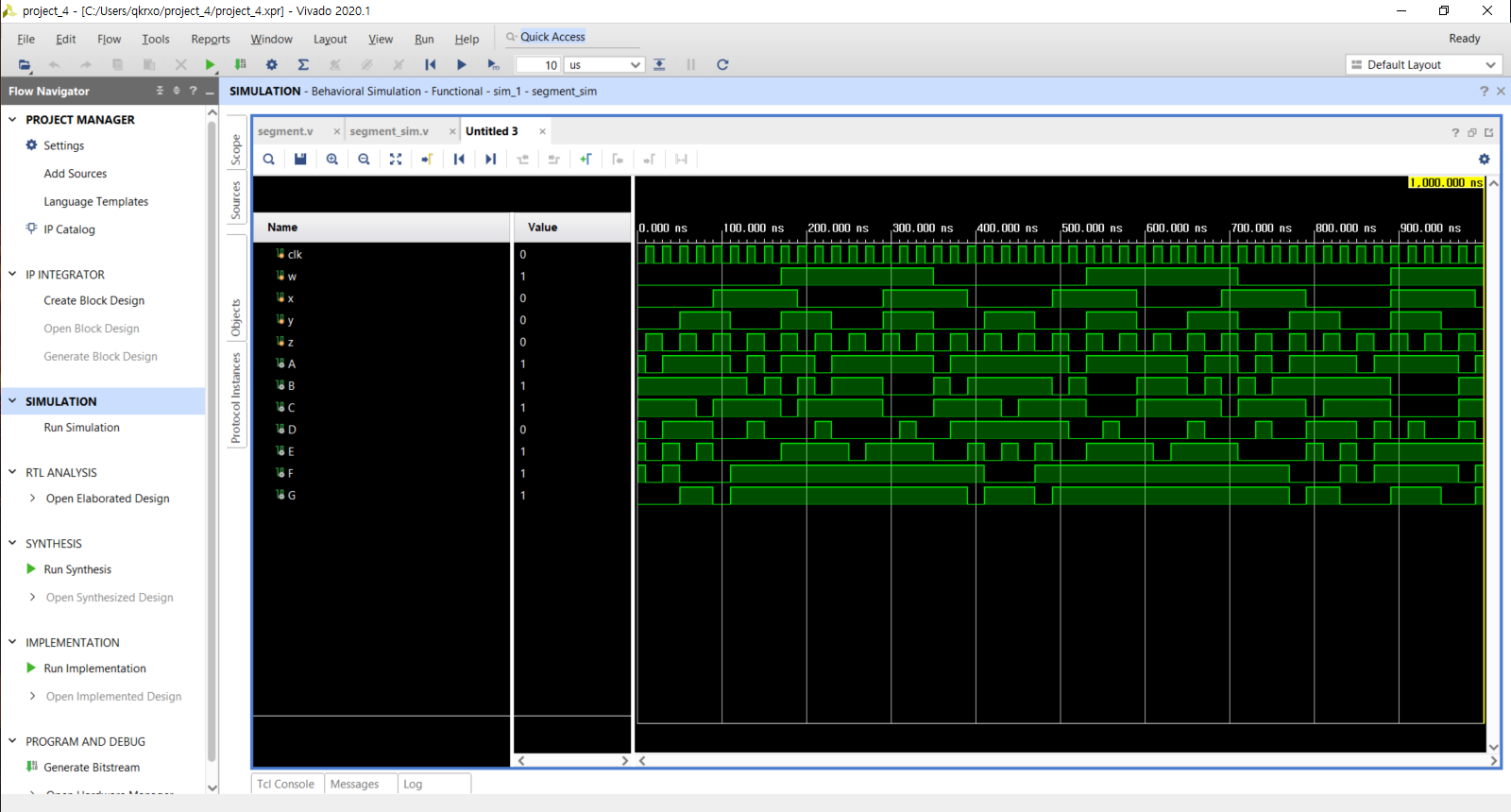
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **wx / yz** | **00** | **01** | **11** | **10** |
| **00** | 0 | 0 | 1 | 1 |
| **01** | 1 | 1 | 0 | 1 |
| **11** | 0 | 1 | 1 | 1 |
| **10** | 1 | 1 | 1 | 1 |

G = wx’ + wy + yz’ + x’y + w’xy’

- 코드

|  |
| --- |
| **segment.v** |
| `timescale 1ns / 1ps  module segment(  input w,x,y,z,  output A,B,C,D,E,F,G,seg  );  assign A = ((~x)&(~z))|((~w)&y)|(x&y)|(w&(~y)&(~z))|(w&(~x)&(~y))|((~w)&x&z);  assign B = ((~x)&(~z))|((~w)&(~x))|((~w)&(~y)&(~z))|((~w)&y&z)|(w&(~y)&z);  assign C = ((~w)&(~y))|((~w)&z)|((~y)&z)|((~w)&x)|(w&(~x));  assign D = ((~w)&(~x)&(~z))|((~w)&(~x)&(~z))|((~x)&y&z)|(x&(~y)&z)|(x&y&(~z));  assign E = ((~x)&(~z))|(y&(~z))|(w&x)|(w&y);  assign F = ((~y)&(~z))|(w&(~x))|(w&y)|((~w)&x&(~y))|(x&(~z));  assign G = (w&(~x))|(w&y)|(y&(~z))|((~x)&y)|((~w)&x&(~y));  assign seg = 1;  endmodule |
| **segment\_sim.v** |
| `timescale 1ns / 1ps  module segment\_sim();  reg clk,w,x,y,z;  wire A,B,C,D,E,F,G;  segment connect(  .w(w), .x(x), .y(y), .z(z),  .A(A), .B(B), .C(C), .D(D), .E(E), .F(F), .G(G)  );  initial begin  clk = 0; w = 0; x = 0; y = 0; z = 0;  end  always clk = #10 ~clk;  always @(posedge clk)begin  z <= ~z;  y <= #40 ~y;  x <= #80 ~x;  w <= #160 ~w;  end  endmodule |

- 시뮬레이션



- FPGA 동작 확인

회로이(가) 표시된 사진

자동 생성된 설명

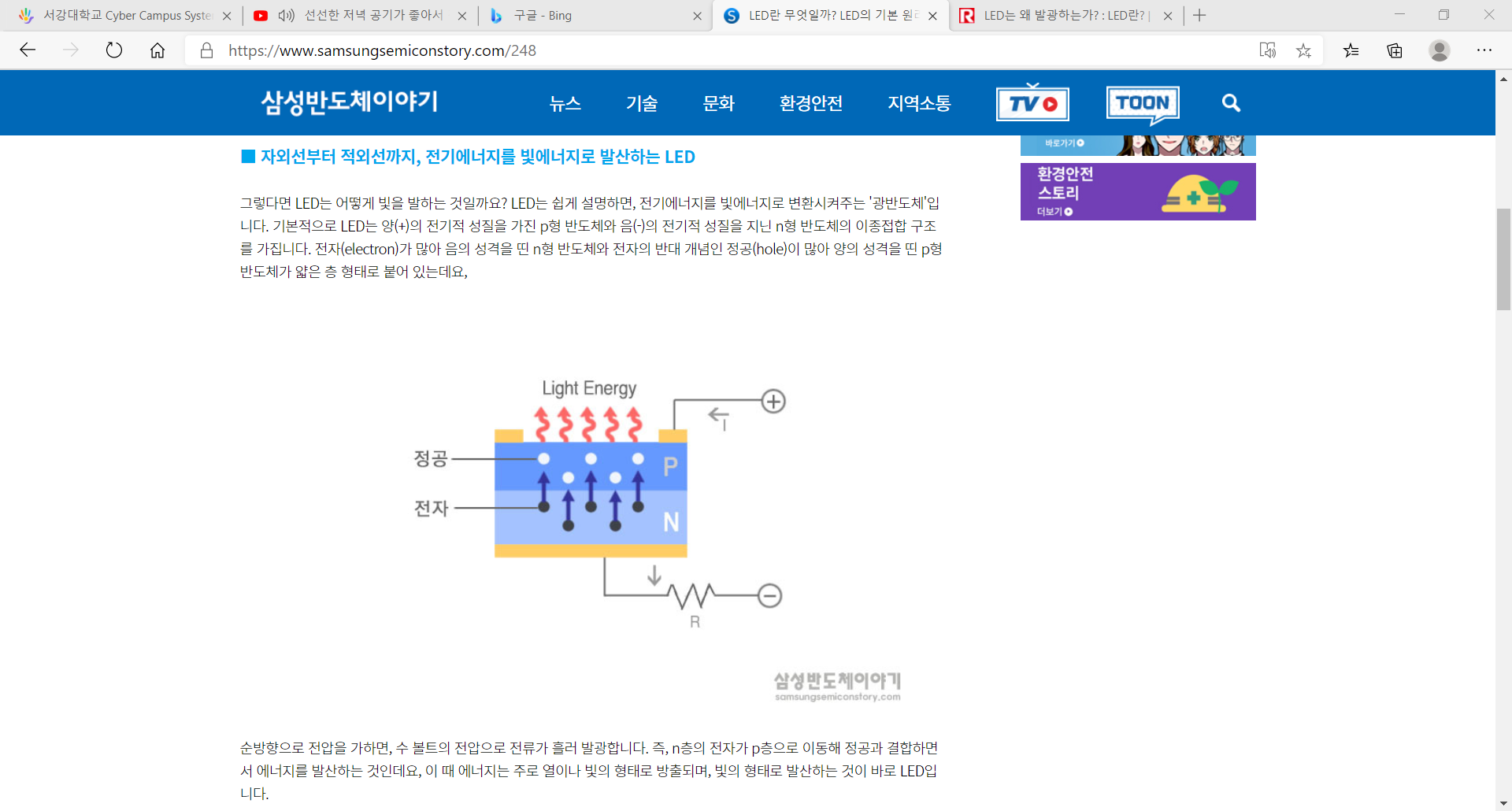
**3. 결과 검토 및 논의 사항.**

7-Segment Display의 진리표를 작성한 후 이를 토대로 카르노맵을 작성하여 output인 7개 LED의 식을 구하였다. Verilog로 이를 구현하여 Simulation을 하였고 FPGA를 사용하여 동작을 확인하였다.

FPGA를 사용하여 실습을 진행하였는데 처음에 코드를 토대로 FPGA를 동작시키는 것에 어려움을 느꼈지만 이번 실습을 통해 FPGA연결 방법과 Vivado프로그램을 사용하여 동작시키는 법을 확인할 수 있었다.

**4. 추가 이론 조사 및 작성**

LED란 Light Emitting Diode의 이니셜을 조합한 것이며 N형 반도체와 P형 반도체를 접합한 광원이다.



여기서 N형 반도체는 전하를 옮기는 운반자로써 자유전자가 사용되는 반도체로 (-)성질을 나타내며 P형 반도체는 전하를 옮기는 운반자로써 N형 반도체와 달리 정공이 사용되는 반도체로 (+)성질을 나타낸다. 반도체에 순방향 전압을 인가하여 전자와 정공이 이동하면서 접합부에서 재결합하고 이를 통해 빛 에너지가 발생하는 원리이다. 백열등과 같은 기존 광원들은 전기 에너지를 열 에너지로 변환하고 이를 빛 에너지로 변환하지만 LED는 전기 에너지를 직접 빛 에너지로 변환하기 때문에 높은 효율을 기대할 수 있다.

LED는 화합물 반도체에 속하는데 화합물 반도체란 하나의 원소로 이루어진 단원소 반도체와 달리, 2종 이상의 원소로 이루어진 반도체이다. LED에서는 어떤 화합물을 쓰는지에 따라 발산하는 빛의 색깔이 달라진다. 즉, 화합물에 조합에 따라 다양한 표현을 할 수 있다.

LED는 전류를 제어하는 저항을 LED와 직렬로 배치하여 다음과 같이 구성할 수 있다.

