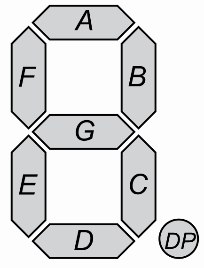
1. 8주차 결과보고서

|  |
| --- |
| * 1. - 7-Segment Display   O 실험 목적  O 결과 및 Simulation 과정  O 결과 검토 및 논의 사항  O 추가 이론 조사 및 작성 |

**20141196 김성희**

1. 7-Segment Display
   1. **가. 실험 목적**
2. 숫자(0~9)와 문자(A~F, A **.** ~ F **.**)에 대한 A~G, DP 진리표 그리기. (숫자는 DP 제외)
3. 숫자(0~9)와 문자(A~F, A **.** ~ F **.**)에 대한 A~G, DP 카르노맵을 그리고 논리식 찾기
4. Verilog를 통해 논리식 표현한 코드 작성 및 시뮬레이션
5. FPGA를 통해 7-Segment Display 구현.

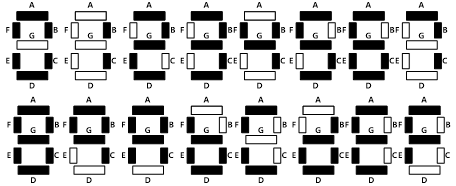
**나. 결과 및 Simulation 과정**

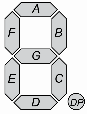
**1. 7-Segment Display 실험 간략 소개**

**왼쪽 그림과 같이 7개의 선과 1개의 점으로 나누어진 display에 불을 켜게 될 부분을 선택해서 숫자와 문자를 나타내는 것을 7-Segment Display라고 한다. 예를들어 A와 B, C를 선택하면 7을 나타내고, A~G를 모두 선택하면 8을 나타낸다.**

**실험에서 각 숫자는 4개의 비트로 표현될 것이며 비트의 각 자리수가 입력 값이 될 것이며 A~G가 출력 값이 될 것이다. (문자(A~F, A . ~ F .)는 3개의 비트로 표현 즉 입력 3개, 출력 8개)**

**다음은 이번 실험에서 표현하게 될 숫자와 문자에 대한 7-Segment Display 표현 그림이다. (DP는 생략하였다.)**

****

**2. 7-Segment Display 진리표**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **숫자** | **입력값**  **abcd** | **출력값** | | | | | | |
| **A** | **B** | **C** | **D** | **E** | **F** | **G** |
| **0** | **0000** | **1** | **1** | **1** | **1** | **1** | **1** | **0** |
| **1** | **0001** | **0** | **1** | **1** | **0** | **0** | **0** | **0** |
| **2** | **0010** | **1** | **1** | **0** | **1** | **1** | **0** | **1** |
| **3** | **0011** | **1** | **1** | **1** | **1** | **0** | **0** | **1** |
| **4** | **0100** | **0** | **1** | **1** | **0** | **0** | **1** | **1** |
| **5** | **0101** | **1** | **0** | **1** | **1** | **0** | **1** | **1** |
| **6** | **0110** | **1** | **0** | **1** | **1** | **1** | **1** | **1** |
| **7** | **0111** | **1** | **1** | **1** | **0** | **0** | **0** | **0** |
| **8** | **1000** | **1** | **1** | **1** | **1** | **1** | **1** | **1** |
| **9** | **1001** | **1** | **1** | **1** | **0** | **0** | **1** | **1** |
| **10** | **1010** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
| **11** | **1011** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
| **12** | **1100** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
| **13** | **1101** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
| **14** | **1110** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |
| **15** | **1111** | **X** | **X** | **X** | **X** | **X** | **X** | **X** |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **문자** | **입력값**  **abcd** | **출력값** | | | | | | | |
| **A** | **B** | **C** | **D** | **E** | **F** | **G** | **dp** |
| **A** | **0000** | **1** | **1** | **1** | **0** | **1** | **1** | **1** | **0** |
| **b** | **0001** | **0** | **0** | **1** | **1** | **1** | **1** | **1** | **0** |
| **C** | **0010** | **1** | **0** | **0** | **1** | **1** | **1** | **0** | **0** |
| **d** | **0011** | **0** | **1** | **1** | **1** | **1** | **0** | **1** | **0** |
| **E** | **0100** | **1** | **0** | **0** | **1** | **1** | **1** | **1** | **0** |
| **F** | **0101** | **1** | **0** | **0** | **0** | **1** | **1** | **1** | **0** |
| **A .** | **0110** | **1** | **1** | **1** | **0** | **1** | **1** | **1** | **1** |
| **b .** | **0111** | **0** | **0** | **1** | **1** | **1** | **1** | **1** | **1** |
| **C .** | **1000** | **1** | **0** | **0** | **1** | **1** | **1** | **0** | **1** |
| **d .** | **1001** | **0** | **1** | **1** | **1** | **1** | **0** | **1** | **1** |
| **E .** | **1010** | **1** | **0** | **0** | **1** | **1** | **1** | **1** | **1** |
| **F .** | **1011** | **1** | **0** | **0** | **0** | **1** | **1** | **1** | **1** |
| **X** | **1100** | **x** | **x** | **x** | **x** | **x** | **x** | **x** | **x** |
| **X** | **1101** | **x** | **x** | **x** | **x** | **x** | **x** | **x** | **x** |
| **X** | **1110** | **x** | **x** | **x** | **x** | **x** | **x** | **x** | **x** |
| **X** | **1111** | **x** | **x** | **x** | **x** | **x** | **x** | **x** | **x** |

**3. 숫자(0~9)와 문자(A~F, A . ~ F .)에 대한 카르노맵과 논리식**

**- 숫자 : 입력: abcd 4개 bit, 출력: A~G**

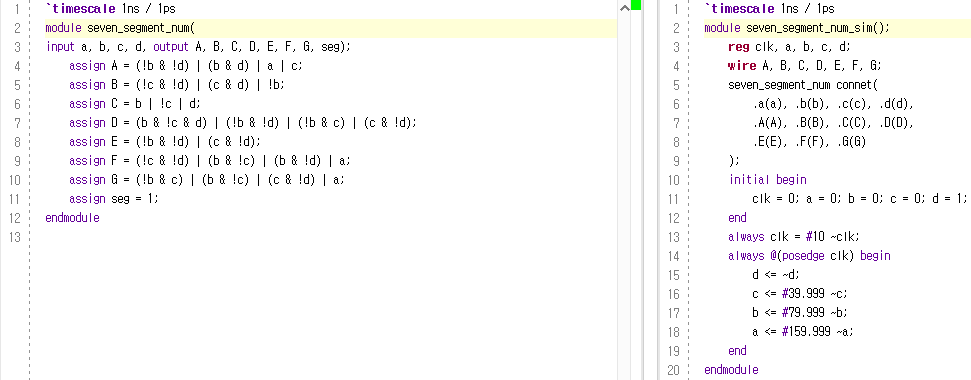
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ab**  **cd** | **00** | **01** | **11** | **10** | | **00** | **1** | **0** | **X** | **1** | | **01** | **0** | **1** | **X** | **1** | | **11** | **1** | **1** | **X** | **X** | | **10** | **1** | **1** | **X** | **X** |   **A =b’d’+bd+a+c** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ab**  **cd** | **00** | **01** | **11** | **10** | | **00** | **1** | **1** | **X** | **1** | | **01** | **1** | **0** | **X** | **1** | | **11** | **1** | **1** | **X** | **X** | | **10** | **1** | **0** | **X** | **X** |   **B =c’d’+cd+b’** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ab**  **cd** | **00** | **01** | **11** | **10** | | **00** | **1** | **1** | **X** | **1** | | **01** | **1** | **1** | **X** | **1** | | **11** | **1** | **1** | **X** | **X** | | **10** | **0** | **1** | **X** | **X** |   **C =b+c’+d** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ab**  **cd** | **00** | **01** | **11** | **10** | | **00** | **1** | **0** | **X** | **1** | | **01** | **0** | **1** | **X** | **0** | | **11** | **1** | **0** | **X** | **X** | | **10** | **1** | **1** | **X** | **X** |   **D =bc’d+b’d’+b’c+cd’** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ab**  **cd** | **00** | **01** | **11** | **10** | | **00** | **1** | **0** | **X** | **1** | | **01** | **0** | **0** | **X** | **0** | | **11** | **0** | **0** | **X** | **X** | | **10** | **1** | **1** | **X** | **X** |   **E =b’d’+cd’** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ab**  **cd** | **00** | **01** | **11** | **10** | | **00** | **1** | **1** | **X** | **1** | | **01** | **0** | **1** | **X** | **1** | | **11** | **0** | **0** | **X** | **X** | | **10** | **0** | **1** | **X** | **X** |   **F =c’d’+bc’+bd’+a** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ab**  **cd** | **00** | **01** | **11** | **10** | | **00** | **0** | **1** | **X** | **1** | | **01** | **0** | **1** | **X** | **1** | | **11** | **1** | **0** | **X** | **X** | | **10** | **1** | **1** | **X** | **X** |   **G =b’c+bc’+cd’+a** |  |

**- 문자 : 입력: abcd 4개 bit, 출력: A~G, DP**

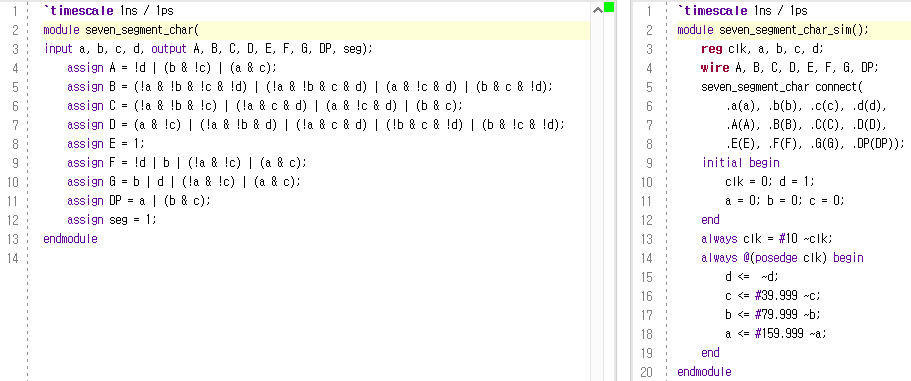
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ab**  **cd** | **00** | **01** | **11** | **10** | | **00** | **1** | **1** | **X** | **1** | | **01** | **0** | **1** | **X** | **0** | | **11** | **0** | **0** | **X** | **1** | | **10** | **1** | **1** | **X** | **1** |   **A =d’+bc’+ac** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ab**  **cd** | **00** | **01** | **11** | **10** | | **00** | **1** | **0** | **X** | **0** | | **01** | **0** | **0** | **X** | **1** | | **11** | **1** | 0 | **X** | **0** | | **10** | **0** | 1 | **X** | **0** |   **B =a’b’c’d’+a’b’cd+ac’d+bcd’** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ab**  **cd** | **00** | **01** | **11** | **10** | | **00** | **1** | **0** | **X** | **0** | | **01** | **1** | **0** | **X** | **1** | | **11** | 1 | **1** | **X** | **0** | | **10** | 0 | **1** | **X** | **0** |   **C =a’b’c’+a’cd+ac’d+bc** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ab**  **cd** | **00** | **01** | **11** | **10** | | **00** | **0** | **1** | **X** | **1** | | **01** | **1** | **0** | **X** | **1** | | **11** | 1 | **1** | **X** | **0** | | **10** | 1 | **0** | **X** | **1** |   **D =ac’+a’b’d+a’cd+b’cd’+bc’d’** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ab**  **cd** | **00** | **01** | **11** | **10** | | **00** | **1** | **1** | **X** | **1** | | **01** | **1** | **1** | **X** | **1** | | **11** | **1** | **1** | **X** | **1** | | **10** | **1** | **1** | **X** | **1** |   **E =1** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ab**  **cd** | **00** | **01** | **11** | **10** | | **00** | **1** | **1** | **X** | **1** | | **01** | **1** | **1** | **X** | **0** | | **11** | **0** | **1** | **X** | **1** | | **10** | **1** | **1** | **X** | **1** |   **F =d’+b+ac+a’c’** |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ab**  **cd** | **00** | **01** | **11** | **10** | | **00** | **1** | **1** | **X** | **0** | | **01** | **1** | **1** | **X** | **1** | | **11** | 1 | **1** | **X** | **1** | | **10** | 0 | **1** | **X** | **1** |   **G =b+d+a’c’+ac** | |  |  |  |  |  | | --- | --- | --- | --- | --- | | **Ab**  **cd** | **00** | **01** | **11** | **10** | | **00** | **0** | **0** | **X** | **1** | | **01** | **0** | **0** | **X** | **1** | | **11** | **0** | **1** | **X** | **1** | | **10** | **0** | **1** | **X** | **1** |   **DP =a+bc** |

**\* bold 사용**

**4. 7-Segment Display Verilog code**

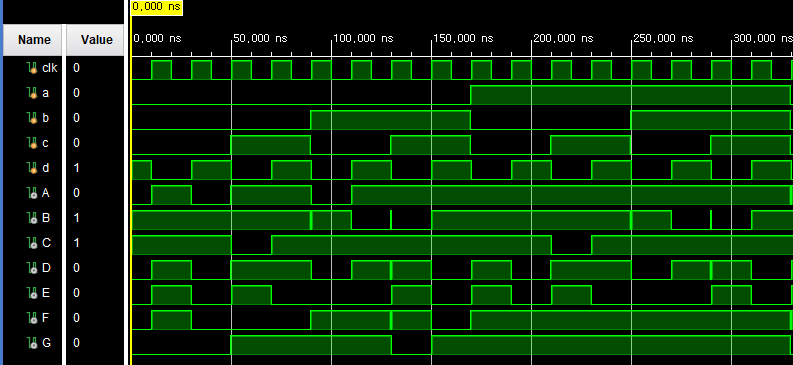
**- 숫자, DesignSource/SimulationSource (seg = segment display on/off 변수 따라서 항상 1) **

**- 문자, DesignSource/SimulationSource**

****

**5. 7-Segment Display Simulation**

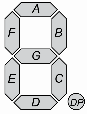
**- 숫자**

****

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |
|  |

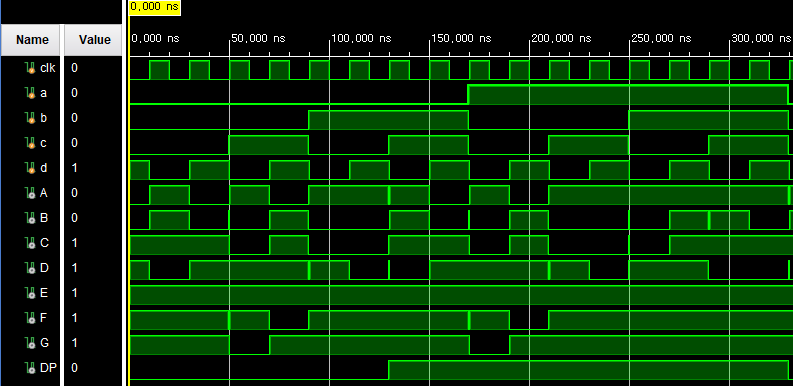
**\* abcd = 0000 -> ABCDEFG=1111110 -> 0표시**

**\* abcd = 0001 -> ABCDEFG=0110000 -> 1표시**

**\* abcd = 0010 -> ABCDEFG=1101101 -> 2표시**

**그 외에 abcd = 0011 ~ 1001까지 각각 3~9표시**

**- 문자**

****

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | |  | |  | | |  | | --- | |  | |  | |
| |  | | --- | |  | |  | | |  | | --- | |  | |  | |

**\* abcd=0000-> ABCDEFG(DP)=11101110-> A표시**

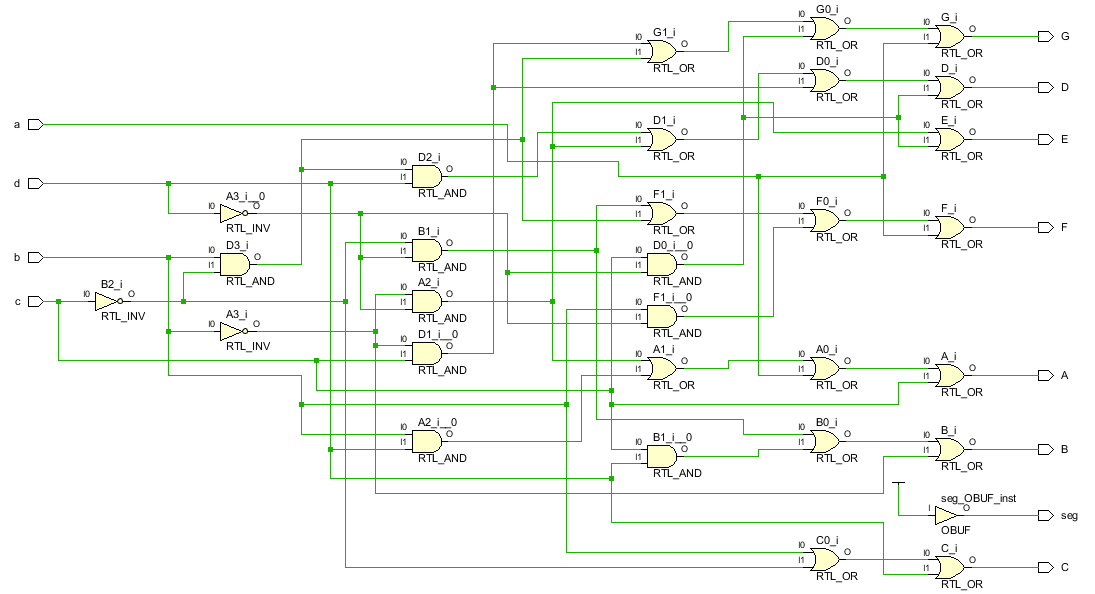
**\* abcd=0110-> ABCDEFG(DP)=11101111-> A .표시**

**\* abcd=0001-> ABCDEFG(DP)=00111110-> b표시**

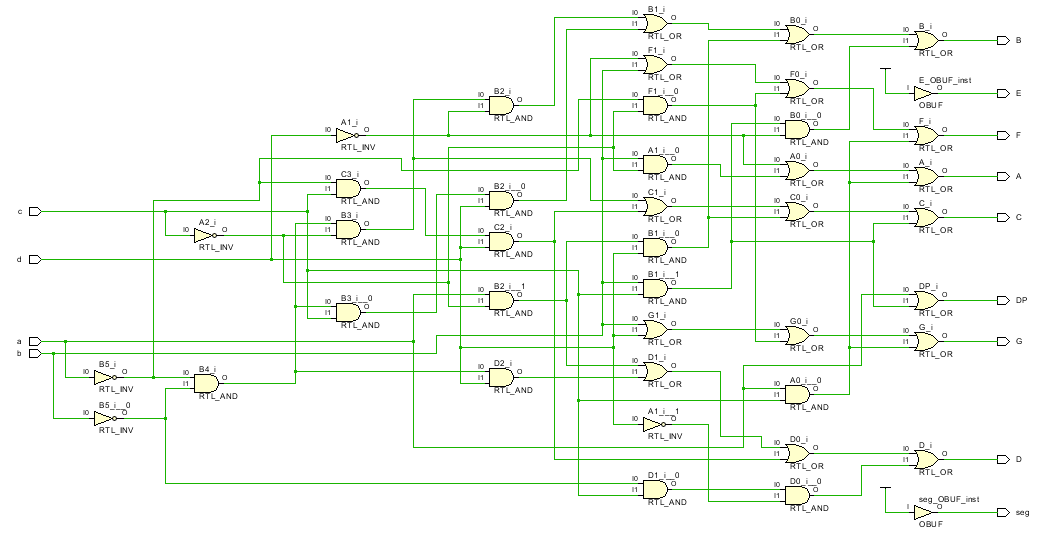
**\* abcd=0111-> ABCDEFG(DP)=00111110-> b .표시**

**6. 7-Segment Display Schematic**

**- 숫자**

****

**- 문자**

****

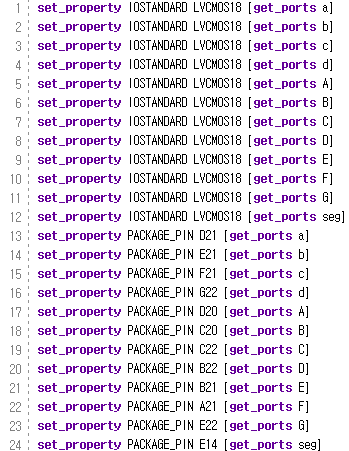
**7. FPGA를 이용한 7-Segment Display 구현**

-Design Sources경로에 code 짜기 -> Run Synthesis & Run Implementation -> Constraints경로에 .xdc 파일 추가하기 -> Open Implemented Design 클릭 후 Constraints Wizard 클릭(Define Target, .xdc파일 타겟 설정) -> Window 탭에서 I/O ports 클릭 -> 원하는 pin 선택 및 LVCMOS18(I/O Std) 선택 -> 저장 후 .xdc reload(아래 그림처럼 코드가 자동으로 짜서 나온다.) -> Generate Bitstream 클릭(Synthesis, Implementation도 자동으로 실행) 후에 Open Hardware Manager 클릭 -> Open Target 클릭 후 Auto Connect 클릭 -> Program Device 클릭 -> FPGA에 업로드 끝

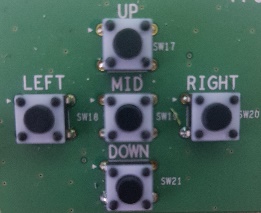
**- 숫자**

**Switch는 LEFT, UP, DOWN, RIGHT를 사용**

**DIGIT는 DIGIT1을 사용**

<- input, ouput pin 배정 (.xdc파일)

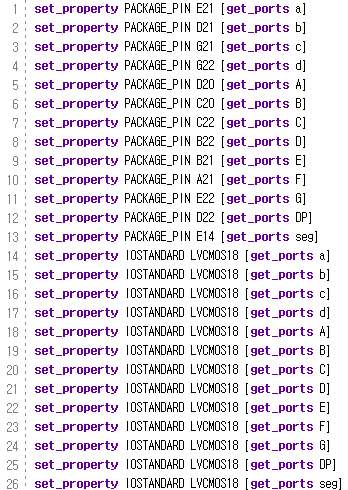
|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| NUMBER |  |  |  |  |  |  |  |  |  |  |
| abcd | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 | 1000 | 1001 |
| PUSH SWITCH | NONE | RIGHT | DOWN | DOWN  RIGHT | UP | UP  RIGHT | UP  DOWN | UP  DOWN  RIGHT | LEFT | LEFT  RIGHT |

 a = LEFT, b = UP, c = DOWN, d = RIGHT (1 -> 0001 -> push RIGHT switch)

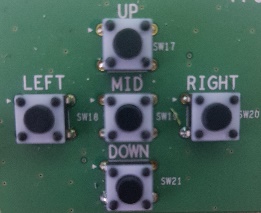
**- 문자**

**Switch는 UP, LEFT, MID, RIGHT를 사용**

**DIGIT는 DIGIT1을 사용**

**** <- input, output pin 배정 ( .xdc파일)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| NUMBER |  |  |  |  |  |  |
| abcd | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 |
| PUSH SWITCH | NONE | RIGHT | MID | MID  RIGHT | LEFT | LEFT  RIGHT |
| NUMBER |  |  |  |  |  |  |
| abcd | 0110 | 0111 | 1000 | 1001 | 1010 | 1011 |
| PUSH SWITCH | LEFT  MID | LEFT  MID  RIGHT | UP | UP  RIGHT | UP  MID | UP  MID  RIGHT |

 a = LEFT, b = UP, c = DOWN, d = RIGHT (1 -> 0001 -> push RIGHT switch)

**다. 결과 검토 및 논의 사항**

1. shcecmatic이 매우 복잡하다. 논리식을 좀 더 줄여서 구현했다면 조금은 덜 복잡했을 것이다.

2. 숫자의 경우 0000~1001 이후의 입력에 대해서 값이 뒤죽박죽이다.

3. 마찬가지로 문자의 경우 0000~1011 이후의 입력에 대해서 값이 제멋대로다.

4. 2, 3번에서 원하는 입력값 이후의 입력에 대한 출력을 막을 수 있는 방안을 마련하는 것이 좋아 보인다. (혹시라도 있을 실수 방지)

**라. 추가 이론 조사 및 작성**

1.999같은 소수를 나타내기 위해서는 0 . 1 . 2 . … 9 . 을 구현해야 한다. 그렇게 되면 입력값이 5개가 필요하다. 카르노맵은 다음과 같이 구하면 된다.

**a=0**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **bc**  **de** | **00** | **01** | **11** | **10** |
| **00** |  |  |  |  |
| **01** |  |  |  |  |
| **11** |  |  |  |  |
| **10** |  |  |  |  |

**a=1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **bc**  **de** | **00** | **01** | **11** | **10** |
| **00** |  |  |  |  |
| **01** |  |  |  |  |
| **11** |  |  |  |  |
| **10** |  |  |  |  |

A~F이외의 알파벳 또한 표현은 가능하나 W, M같은 경우는 모양이 이상해지고 X또한 H와 같은 모양을 쓴다. 따라서 약속을 하기 전에는 구분을 못하게 될 수도 있다.

순서대로

A b C d E F G h i J K L M

n o P q r S t U v W X y Z (S와 Z는 숫자 5, 2와 구분이 안 간다)

