|  |  |  |
| --- | --- | --- |
| 디지털 시스템 설계 및 실험 결과보고서 | | |
| 이름 : 손명준 | | 학번 : 2018170914 |
| **실험제목** | 12 : Keypad 스캔회로 설계 | |
| **실험목표** | 1. Keypad scan 회로를 설계한다. 2. Keypad로 입력받은 값을 7-segment array에 띄운다. | |
| **실험결과** | | |
| **1. Keypad scan 회로 설계**  Keypad scan 회로는 다음과 같다.    버튼이 12개 있는 keypad로부터 어떤 키가 입력되었는지에 대한 정보인 Keypad\_in 신호를 받아 scan\_out 신호와 valid 신호를 출력한다. keypad\_in 신호가 몇 클럭 이상 연속으로 지속되어도 scan\_out 신호와 valid 신호는 1 클럭 동안만 유효하다.  이번 실험에서는 7-segment array의 모든 7-segment를 활용하였다. 입력 위치를 변경하기 위해 #버튼을 자리올림 버튼으로 사용하였다. 8개의 모든 7-seg를 사용하기 위해 키패드로 입력된 정보를 올바른 위치의 7-segment와 연결하는 동작을 위한 모듈이 필요하다. 해당 모듈은 입력된 신호에 대해 올바른 7-segment 출력 신호를 생성하고, 출력 위치에 맞는 register에 해당 신호를 전달한다. register는 전달받은 신호를 저장하고, 저장된 신호를 7-segment controller 모듈로 전달한다. 7-segment controller 모듈은 순차적으로 7-segment array에 신호를 보내 지금까지 입력된 모든 숫자가 적절한 위치의 7-segment에 출력되도록 통제한다.  display.v의 모듈은 키패드로 입력된 신호를 7-segment 출력 신호로 변환하여 적절한 register에 전달하는 역할을 수행한다, 12’b00…001부터 12’b10…000까지 12개의 입력 경우에 대해 7-segment 출력 신호로 변환하는 과정을 진행한다. \* 버튼과 # 버튼은 7-segment로 숫자를 출력되지 않으며 특수한 기능을 수행한다. \*버튼은 원래 제공된 코드에서는 키패드 입력을 시작하도록 하는 enabler로 설정되었지만, 실험의 편의를 위해 처음부터 en을 1로 설정한 이후에는 필요가 없어져 다른 기능으로 바꾸었다. # 버튼은 출력되는 자리를 한 단계 오른쪽으로 움직이는 동작을 수행한다. 숫자가 출력될 자리는 r9 변수를 이용해 설정한다. #버튼이 입력될 때 마다 r9를 1씩 증가시킨다. 8번째 7-segment까지 입력한 후 # 버튼을 누르면 overflow로 인해 첫 번째 7-segement로 돌아가는 것을 확인하였다.    각 keypad 입력 신호에 대해 7-segment 출력 신호로 변환하는 코드를 완성하였다. # 버튼을 눌러 자리를 옮긴 후 바로 0이 출력되도록 하기 위해 12’b10…000 신호에 대해 w에 0의 7-segment 출력 신호를 대입하는 코드를 추가하였다.  \*버튼은 7-segment의 on/off 버튼으로 용도를 바꾸어 보았다. 기존의 register 모듈에서는 한 번 결정된 out 신호는 변경되지 않았다. on off 기능을 구현하기 위해 register 모듈에서 en 신호에 따라 out 신호가 바뀌도록 코드를 수정하였다.    En이 0으로 변하면 해당 7-segment를 끄는 signal을 out에 연결하였다. en이 다시 1로 변하면 이미 저장된 data 변수에서 해당 register가 가지고 있던 신호를 다시 불러오도록 하였다. 다시 생각해보니 어차피 display 모듈에서 해당 register로 동일한 신호를 꾸준히 출력하고 있기 때문에 굳이 data로 저장할 필요는 없었다고 생각된다.    # 버튼을 누르면 다음 위치로 커서가 이동한다.    # 버튼을 포함한 모든 버튼의 입력을 올바르게 처리하였다.    8개의 모든 7-segment에서 정상적으로 출력되는 것을 확인하였다.  Testbench를 통해 살펴본 파형은 아래와 같다.  Reset, press 2      Press #      Press 0      Press #      Press 0      Press #      Press 1      Press \* | | |
| **토의** |  | |
|  | | |