Logic Design Lab Report: Week 5

2013-11826 임주경

1. Introduction

Verilog의 개념을 이해하고, 7-segment decoder를 behavioral description으로 구현하여, 직접 테스트 해본다.

2. Implementation

Verilog 모듈을 이용해서 behavioral description 표현을 사용하여 Decoder를 구현한다. 구현하고자 하는 것은 Klingon number system으로, 4-bit의 Input을 받으면, 아래 그림 1에 맞게 Input의 decimal number에 맞게 7-bit의 Output이 출력되는 7-segment decoder이다. 이때, 각 Output의 LSB은 그림 2의 G, MSB는 그림 2의 A에 해당한다. Input이 9~15의 값을 가질 때는 Output은 7'b0000000 값이 출력되도록 구현한다.

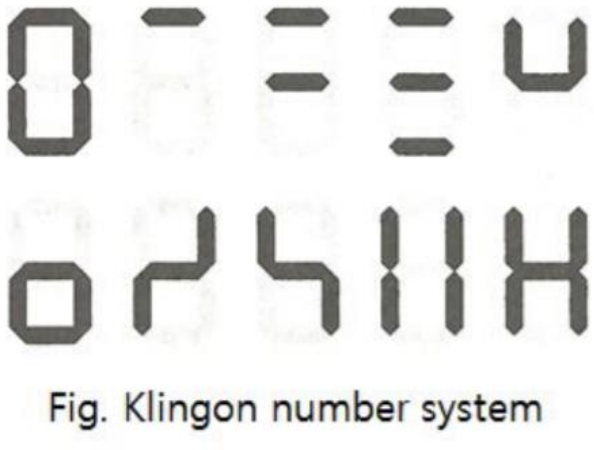

그림1. Klingon number system

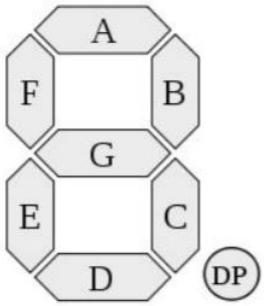

그림2. Output form

3. Result

Input I는 [3:0] 벡터로, Output O는 [6:0] 벡터로 표현한다. 프로그래밍은 behavioral description 표현을 사용한다. 그러므로, Output 7-bit 벡터를 register로 지정하고, always 구문에서 Input에 대한 case 문을 사용하였다. 각 케이스는 Input이 0~9까지일 때, 결과에 맞는 Output값을 입력했고, 나머지 경우에 대해서는 default 처리를 하여 7'b0000000이 출력되도록 하였다. 테스트는 Input 0~15 까지 16세트 진행하였으며, delay unit time은 20으로 설정하였다. 결과 waveform은 그림 3

과 같다.

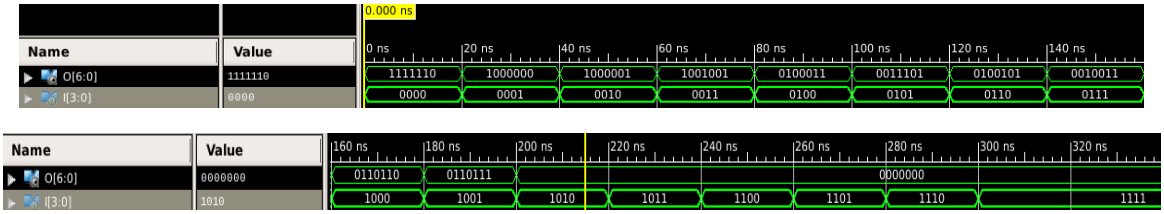

그림3. Simulation 결과 waveform

Input (Decimal)	Output (7'b)
0	1111110
1	1000000
2	1000001
3	1001001
4	0100011
5	0011101
6	0100101
7	0010011
8	0110110
9	0110111
10	0000000
11	0000000
12	0000000
13	0000000
14	0000000
15	0000000

표1. Klingon number system truth table

4. Conclusion/Discussion

Case 구문을 사용해서 어렵지 않게 Klingon number system 7-segment decoder를 구현할 수 있었다. 주의해야할 점은 behavioral description을 사용하였기 때문에, always에서 Output에 assign 해줄 때는, reg 명령어를 사용해서 Output을 선언해주어야 한다는 것이다. 또한, 벡터를 표현할 때는 7'b0011010 과 같이 형식을 잘 지켜주어야 한다. 그 결과, 시뮬레이션의 waveform은 표1의 truth table과 정확히 같은 결과를 나타냄을 확인할 수 있었다.