

0. Discussion

(1) Concepts that I learned during the lab session

지난 주에 배웠던 세 가지의 verilog 프로그래밍 방법을 다시 한 번 복습할 수 있었다. Verilog 프로그래밍 방법은 Structural Description 방법, Data Flow Description 방법, Behavioral Description 방법이 있다. 지난 주 랩 보고서를 작성하는 과정에서 각 방법의 장단점을 조사했었는데, 이번 주 랩 세션에서 BCD to 7 segments와 Klingon number system을 세 가지 방법으로 직접 구현하는 과정에서 각각의 장단점을 더 깊이 이해할 수 있었다.

Structural Description 방법은 하드웨어 컴포넌트 간의 연결을 명확하게 나타내지만 구현하는데 오랜 시간이 걸리며, Data Flow Description 방법은 Structural Description 방법에 비해 구현 속도가 빠르지만 기저 구조에 대한 정보를 얻기 힘들다. 마지막으로, Behavioral Description 방법은 유연성이 높다는 장점이 있지만 Structural Description 방법과 마찬가지로 하드웨어의 기저 구조에 대한 구체적인 정보를 얻기 힘들다는 단점이 있다.

(2) Any errors that I made

이번 랩 세션에는 BCD to 7 segment와 Klingon number system을 각각 Structural Description, Data Flow Description, Behavioral Description으로 구현해야 했다. 비교적 구현이 쉬운 Data Flow Description과 Behavioral Description에 대해 구현 및 테스트까지 우선 마무리하고 팀원들과 역할을 분담하여 BCD to 7 segment와 Klingon number system의 Structural Description을 구현하였었다. Klingon number system의 진리표를 그리고 K-map을 통해 식을 minimize 한 후 Structural Description 방법으로 구현을 마쳤지만 테스트 코드가 올바르게 작성되었음에도 불구하고 시뮬레이션 결과 Klingon number system에서 3으로 표시되어야 하는 부분이 2로 표시가 되는 문제가 있었다.

BCD to 7 segment를 Structural Description 방법으로 구현하기 위해 다른 팀원이 K-map을 그렸고 이를 minimize 하는 과정에서 on-set을 실수로 off-set으로 취급하였다. 랩 세션이 끝나고 각자 보고서를 작성하는 과정에서 해당 오류를 발견하여 수정하였다.

(3) How to correct my errors

Klingon number system에서 output 3에 대한 K-map을 다시 살펴보니 한 개의 on-set을 실수로 놓쳤다는 것을 확인하였다. 기존의 식에 한 개의 prime implicant를 더해서 식을 고쳤고, 베릴로그

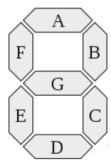
코드 내에서도 수정해주니 decoder가 정상적으로 동작하는 것을 확인할 수 있었다.

팀원들의 노트북에 Xilinx가 안깔려서 월요일에 302동 310-2호로 출석하여 Structural Description 방법으로 구현한 BCD to 7 segment 코드를 수정해주었다. 그 후 시뮬레이션 결과를 확인해보니 decoder가 정상적으로 동작하는 것을 확인할 수 있었다.

1. Lab Practice

(1) Make a truth table with BCD to 7-segment outputs

아래의 이미지는 7-segment 디스플레이에서 각각의 부분을 A~G 까지의 알파벳으로 나타낸 것이다. Input은 4자리의 이진수이고, 아웃풋은 A segment ~ G segment이다.



아래의 표는 BCD to 7-segment outputs을 진리표로 나타낸 것이다. Input을 각각 X, Y, Z, W으로 표기하였는데, X가 MSB, W가 LSB를 의미한다. 또한, 아웃풋을 각각 A부터 G까지의 알파벳으로 표기하였는데, 순서대로 위 디스플레이 이미지의 A segment ~ G segment를 의미한다.

X	Y	Z	W	A	B	C	D	E	F	G
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	1	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	-	-	-	-	-	-	-	-	-
1	1	-	-	-	-	-	-	-	-	-

(2) Make Boolean expressions according to the table

위의 진리표를 참고하면 각각의 아웃풋에 대해 Boolean expressions를 구할 수 있다.

$$A = X'Y'Z'W' + X'Y'ZW' + X'Y'ZW + X'YZ'W + X'YZW + X'YZW' + XY'Z'W' + XY'Z'W$$

$$B = X'Y'Z'W' + X'Y'Z'W + X'Y'ZW' + X'Y'ZW + X'YZ'W' + X'YZW + XY'Z'W' + XY'Z'W$$

$$C = X'Y'Z'W' + X'Y'Z'W + X'Y'ZW + X'YZ'W' + X'YZ'W + X'YZW' + X'YZW + XY'Z'W' + XY'Z'W$$

$$D = X'Y'Z'W' + X'Y'ZW' + X'Y'ZW + X'YZ'W + X'YZW' + XY'Z'W' + XY'Z'W$$

$$E = X'Y'Z'W' + X'Y'ZW' + X'YZW' + XY'Z'W'$$

$$F = X'Y'Z'W' + X'YZ'W' + X'YZ'W + X'YZW' + X'YZW + XY'Z'W' + XY'Z'W$$

$$G = X'Y'ZW' + X'Y'ZW + X'YZ'W' + X'YZ'W + X'YZW' + XY'Z'W' + XY'Z'W$$

(3) Minimize them

K-map을 이용하면 Don't care 조건을 활용하여 효과적으로 minimize 할 수 있다. 각각의 output A ~ G에 대하여 K-map을 그리고 minimize하면 다음과 같다.

$$A = X + Z + YW + Y'W'$$

ZW \ XY	00	01	11	10
00	1	0	X	1
01	0	1	X	1
11	1	1	X	X
10	1	1	X	X

$$B = Y' + Z'W' + ZW$$

ZW \ XY	00	01	11	10
00	1	1	X	1
01	1	0	X	1
11	1	1	X	X
10	1	0	X	X

$$C = Y + Z' + W$$

ZW \ XY	00	01	11	10
00	1	1	X	1
01	1	1	X	1
11	1	1	X	X
10	0	1	X	X

$$D = X + ZW' + Y'Z + YZ'W + Y'W'$$

ZW W XY	00	01	11	10
00	1	0	X	1
01	0	1	X	1
11	1	0	X	X
10	1	1	X	X

$$E = ZW' + Y'W'$$

ZW W XY	00	01	11	10
00	1	0	X	1
01	0	0	X	0
11	0	0	X	X
10	1	1	X	X

$$F = Z'W' + X + Y$$

ZW W XY	00	01	11	10
00	1	1	X	1
01	0	1	X	1
11	0	1	X	X
10	0	1	X	X

$$G = X + YZ' + ZW' + Y'Z$$

ZW W XY	00	01	11	10
00	0	1	X	1
01	0	1	X	1
11	1	0	X	X
10	1	1	X	X

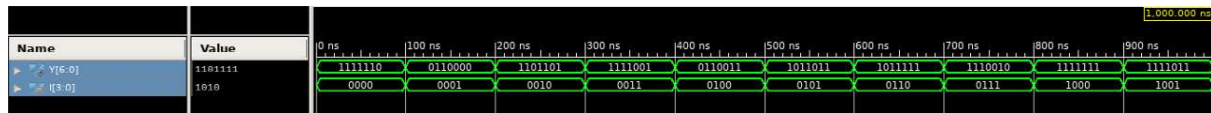
(4) Implement a decoder according to the expressions

Structural Description, Data Flow Description, Behavioral Description 방법으로 decoder를 각각 구현하고 테스트를 진행한 결과는 아래와 같다. Structural Description은 Schematic을, Data Flow Description은 Truth table을 대체하는 방법이고 Behavioral Description은 'How'에 초점을 맞춘 방법임에 유의하여 각각의 방법으로 decoder를 구현한 코드는 압축 파일 내에서 본 레포트와 동일

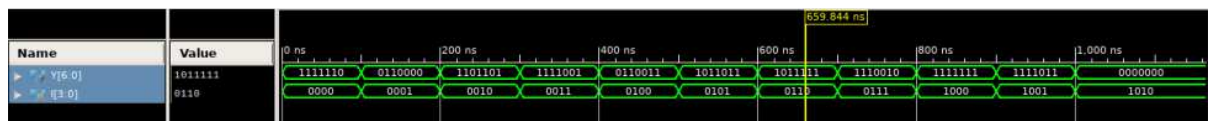
한 위치에 있다. (decoder 구현 파일 : 'bcd_structural.v', 'bcd_dataflow.v', 'bcd_behavioral.v', 테스트 코드 : 'bcd_structural_test.v', 'bcd_dataflow_test.v', 'bcd_behavioral_test.v')

시뮬레이션 결과 세 가지 방법으로 구현한 decoder가 모두 잘 작동하는 것을 확인할 수 있다.

- Structural Description



- Data Flow Description



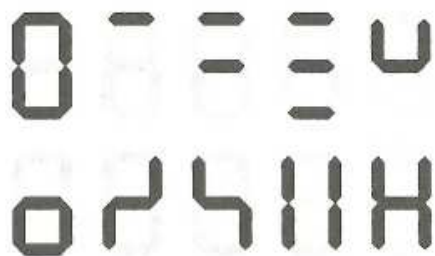
- Behavioral Description



2. Homework

(1) Make a truth table with Klingon number system

아래의 이미지는 0부터 9까지의 숫자를 Klingon number system으로 표현한 것이다. 앞에서와 같이 7-segment 디스플레이에서 각각의 부분을 A~G 까지의 알파벳으로 나타내자. Input은 4자리 의 이진수이고, 아웃풋은 A segment ~ G segment이다.



아래의 표는 BCD to 7-segment outputs을 진리표로 나타낸 것이다. Input을 각각 X, Y, Z, W으로 표기하였는데, X가 MSB, W가 LSB를 의미한다. 또한, 아웃풋을 각각 A부터 G까지의 알파벳으로 표기하였는데, 순서대로 7 segment display의 A segment ~ G segment를 의미한다.

X	Y	Z	W	A	B	C	D	E	F	G
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	1
0	0	1	1	1	0	0	1	0	0	1
0	1	0	0	0	1	0	0	0	1	1
0	1	0	1	0	0	1	1	1	0	1
0	1	1	0	0	1	0	0	1	0	1
0	1	1	1	0	0	1	0	0	1	1
1	0	0	0	0	1	1	0	1	1	0
1	0	0	1	0	1	1	0	1	1	1
1	0	-	-	-	-	-	-	-	-	-
1	1	-	-	-	-	-	-	-	-	-

(2) Make a boolean expressions according to the table

위의 진리표를 참고하면 각각의 아웃풋에 대해 Boolean expressions를 구할 수 있다.

$$A = X'Y'Z'W' + X'Y'Z'W + X'Y'ZW' + X'Y'ZW$$

$$B = X'Y'Z'W' + X'YZ'W' + X'YZW' + XY'Z'W' + XY'Z'W$$

$$C = X'Y'Z'W' + X'YZ'W + X'YZW + XY'Z'W' + XY'Z'W$$

$$D = X'Y'Z'W' + X'Y'ZW + X'YZ'W$$

$$E = X'Y'Z'W' + X'YZ'W + X'YZW' + XY'Z'W' + XY'Z'W$$

$$F = X'Y'Z'W' + X'YZ'W' + X'YZW + XY'Z'W' + XY'Z'W$$

$$G = X'Y'ZW' + X'Y'ZW + X'YZ'W' + X'YZ'W + X'YZW' + X'YZW + XY'Z'W$$

(3) Minimize them

K-map을 이용하면 Don't care 조건을 활용하여 효과적으로 minimize 할 수 있다. 각각의 output A ~ G에 대하여 K-map을 그리고 minimize하면 다음과 같다.

$$A = X'Y'$$

ZW \ XY	00	01	11	10
00	1	0	X	0
01	1	0	X	0

11	1	0	X	X
10	1	0	X	X

$$B = X + Z'W' + YW'$$

ZW W XY	00	01	11	10
00	1	1	X	1
01	0	0	X	1
11	0	0	X	X
10	0	1	X	X

$$C = X + YW + Y'Z'W'$$

ZW W XY	00	01	11	10
00	1	0	X	1
01	0	1	X	1
11	0	1	X	X
10	0	0	X	X

$$D = X'Y'Z'W' + YZ'W + Y'ZW$$

ZW W XY	00	01	11	10
00	1	0	X	0
01	0	1	X	0
11	1	0	X	X
10	0	0	X	X

$$E = X + Y'Z'W' + YZW' + YZ'W$$

ZW W XY	00	01	11	10
00	1	0	X	1
01	0	1	X	1
11	0	0	X	X
10	0	1	X	X

$$F = X + Z'W' + YZW$$

ZW W XY	00	01	11	10
----------------	-----------	-----------	-----------	-----------

00	1	1	X	1
01	0	0	X	1
11	0	1	X	X
10	0	0	X	X

$$G = Y + Z + XW$$

ZW W XY	00	01	11	10
00	0	1	X	0
01	0	1	X	1
11	1	1	X	X
10	1	1	X	X

(4) Implement a decoder according to the expressions

Structural Description, Data Flow Description, Behavioral Description 방법으로 decoder를 각각 구현하고 테스트를 진행한 결과는 아래와 같다. Structural Description은 Schematic을, Data Flow Description은 Truth table을 대체하는 방법이고 Behavioral Description은 'How'에 초점을 맞춘 방법임에 유의하여 각각의 방법으로 decoder를 구현한 코드는 압축 파일 내에서 본 레포트와 동일한 위치에 있다. (decoder 구현 파일 : 'klington_structural.v', 'klington_dataflow.v', 'klington_behavioral.v' 테스트 코드 : 'klington_structural_test.v', 'klington_dataflow_test.v', 'klington_behavioral_test.v')

시뮬레이션 결과 세 가지 방법으로 구현한 decoder가 모두 잘 작동하는 것을 확인할 수 있다.

- Structural Description



- Data Flow Description



- Behavioral Description

