

MUX& Decoder 구현

상명대학교 휴먼지능정보공학과

201810776 소재휘

Lab 03-1. Design with MUX

- Consider the function $F(A,B,C,D) = \sum m(0,2,3,8,10,11,12,14,15)$.
Design F with a 8-to-1 MUX
 - Design a 8-to-1 MUX with Enable (with VHDL)
 - Verify the 8-to-1 MUX with simulation
 - Design a circuit that produce F, by using the MUX you just made
 - Verify with simulation the circuit
 - Discuss the result

1. Truth-table

$$F(A,B,C,D) = \sum m(0,2,3,8,10,11,12,14,15)$$

MSB를 A로 잡고

$F(A,B,C,D) = \sum m(0,2,3,8,10,11,12,14,15)$ 의
진리표는 다음과 같다.

A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

2. VHDL Capture

8-to-1 MUX

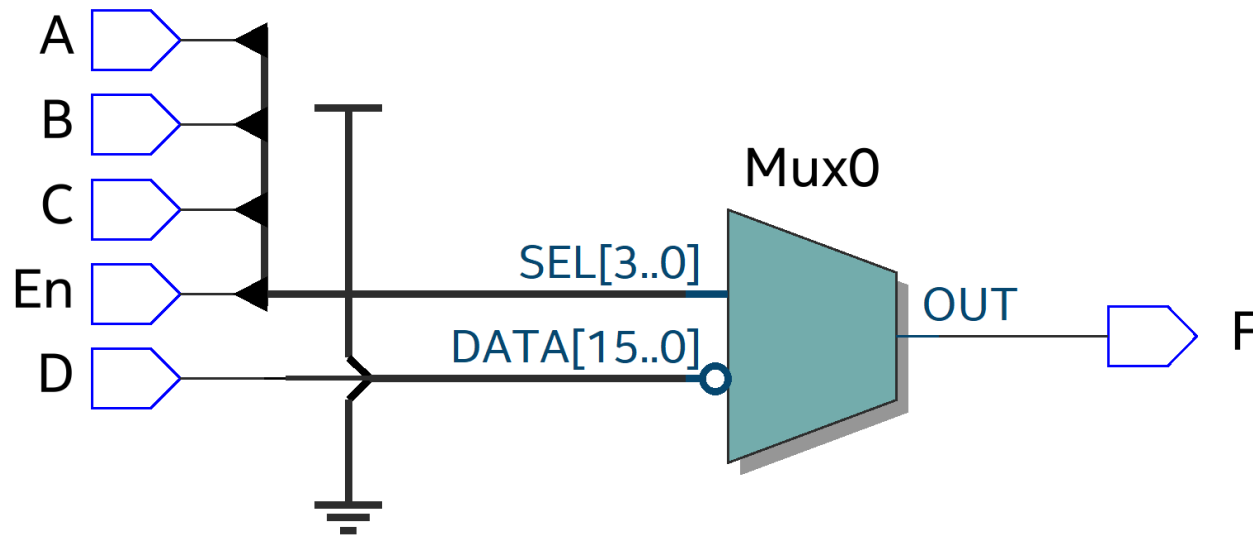
8-to-1 MUX 를 구현하기
위해 Enable과 Select
A,B,C에 따라서 F의 값을
결정하는 VHDL 코드를
구현했다.

```
LIBRARY ieee ;
USE ieee.std_logic_1164.all ;

ENTITY mux8to1 IS
    PORT ( A,B,C,D : in std_logic ;
           En : In std_logic;
           F : OUT STD_LOGIC ) ;
END mux8to1 ;

ARCHITECTURE Behavior OF mux8to1 IS
    signal Ens : std_logic_vector(3 downto 0);
BEGIN
    Ens <= En & A & B & C;
    WITH Ens SELECT
        F <= not D WHEN "1000",
            '1' WHEN "1001",
            '0' WHEN "1010",
            '0' when "1011",
            not D WHEN "1100",
            '1' WHEN "1101",
            not D WHEN "1110",
            '1' when "1111",
            '0' when OTHERS;
END Behavior ;
```

3. RTL viewer capture

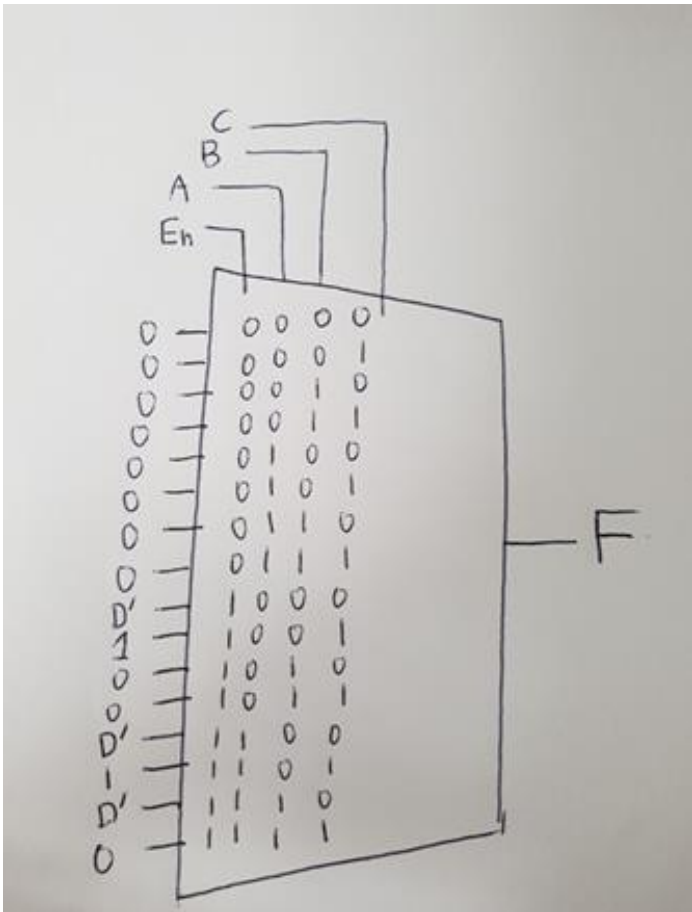


$$F(A,B,C,D) = \sum m(0,2,3,8,10,11,12,14,15).$$

Design F with a 8-to-1 MUX

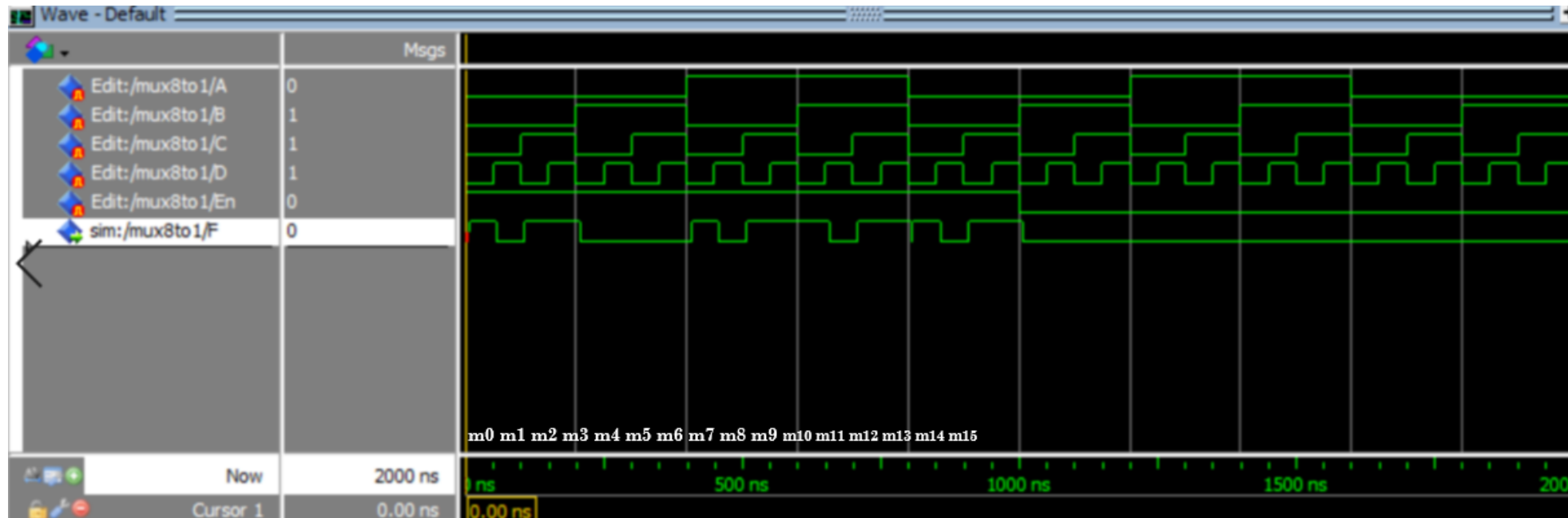
A, B, C, Enable이 Select 포트, Select에 따라 D(0)부터 D(15)까지의 값이 F로 Output 되는 MUX 회로가 잘 구현되었음을 확인할 수 있었다.

4. Circuit Design



진리표를 MUX회로로 구현하면 다음과 같다.

5. Simulation capture



8-to-1 MUX를 시뮬레이팅 해 보았다. $F(A,B,C,D) = \sum m(0,2,3,8,10,11,12,14,15)$ 의 결과와 같이 F의 값이 0과 1로 결정되었다. 예를 들어 진리표에서 A,B,C,D가 0010 일 때 F가 1이었다. 위의 그림에서 0010이 해당하는 m2의 F값이 1로 결정된 것을 확인할 수 있었다. 또한 Enable이 0일 때에는 F의 값이 무조건 0으로 Output되는 것을 확인할 수 있었다.

Lab 03-2. Design with Decoder

- Consider the function $F(A,B,C,D) = \sum m(0,2,3,8,10,11,12,14,15)$.
Design F with several 4-to-16 Decoder.
 - Design a 4-to-16 Decoder with Enable (with VHDL)
 - Verify the 4-to-16 Decoder with simulation
 - Design a circuit that produce F, by using the Decoder you just made
 - Verify with simulation the circuit
 - Discuss the result

1. Truth-table

$$F(A,B,C,D) = \sum m(0,2,3,8,10,11,12,14,15)$$

MSB를 A로 잡고

$F(A,B,C) = \sum m(0,2,3,8,10,11,12,14,15)$ 의
진리표는 다음과 같다.

A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

2. VHDL

Capture

4-to-16 Decoder

4-to-16 Decoder를 구현하기 위해 Enable과 A,B,C,D에 따라 Y(0)부터 Y(15)까지의 값을 16 비트로 output 되도록 설계하였고 진리표에서 1로 나오게 되는 y의 값을 or로 묶어 F로 output되도록 하였다.

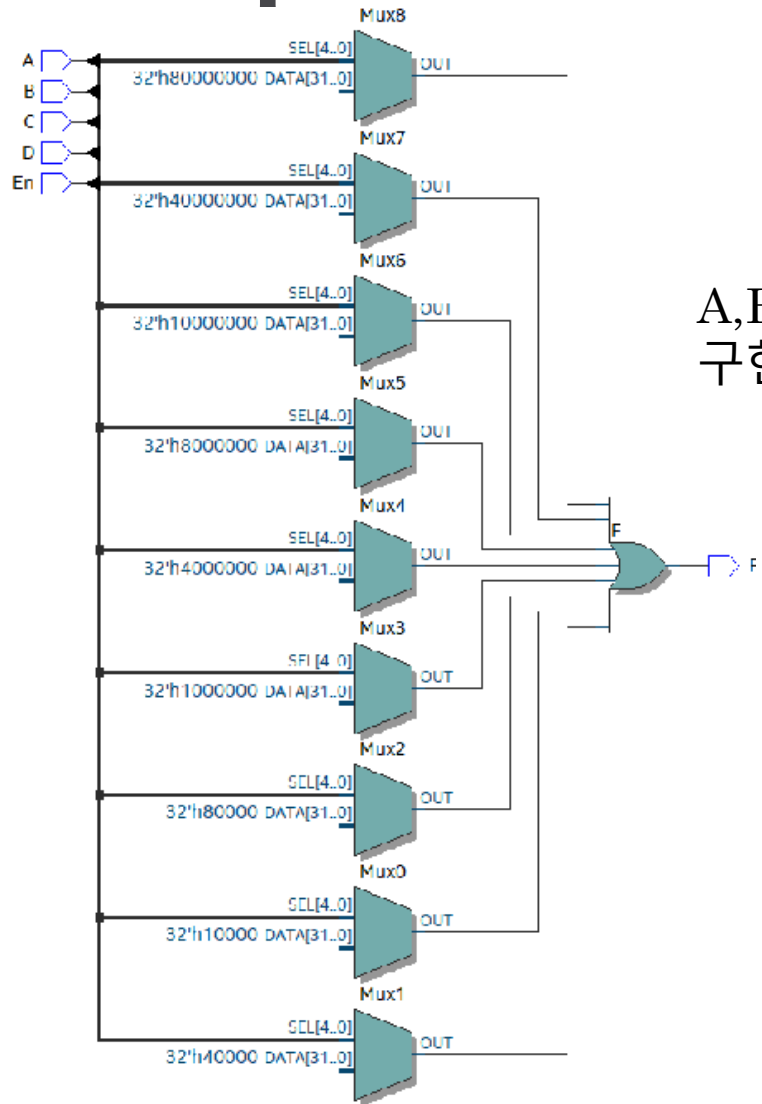
```
BEGIN
    Enw <= En & A & B & C & D ;

    WITH Enw SELECT
        Y <= "1000000000000000" WHEN "10000",
            "0100000000000000" WHEN "10001",
            "0010000000000000" WHEN "10010",
            "0001000000000000" WHEN "10011",
            "0000100000000000" WHEN "10100",
            "0000010000000000" WHEN "10101",
            "0000001000000000" WHEN "10110",
            "0000000100000000" WHEN "10111",
            "0000000010000000" WHEN "11000",
            "0000000001000000" WHEN "11001",
            "0000000000100000" WHEN "11010",
            "0000000000010000" WHEN "11011",
            "0000000000001000" WHEN "11100",
            "0000000000000100" WHEN "11101",
            "0000000000000010" WHEN "11110",
            "0000000000000001" WHEN "11111",
            "0000000000000000" when OTHERS;

    F <= Y(0) or Y(2) or Y(3) or Y(8) or Y(10) or Y(11) or Y(12) or Y(14) or Y(15);

END Behavior ;
```

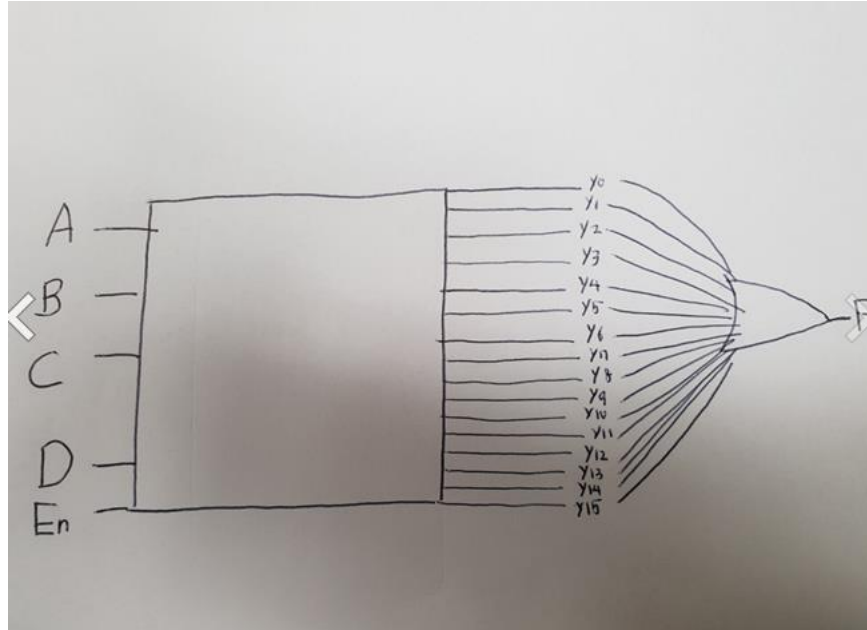
3. RTL viewer capture



$F(A,B,C,D) = \sum m(0,2,3,8,10,11,12,14,15).$
 Design F with a 4-to-16 Decoder

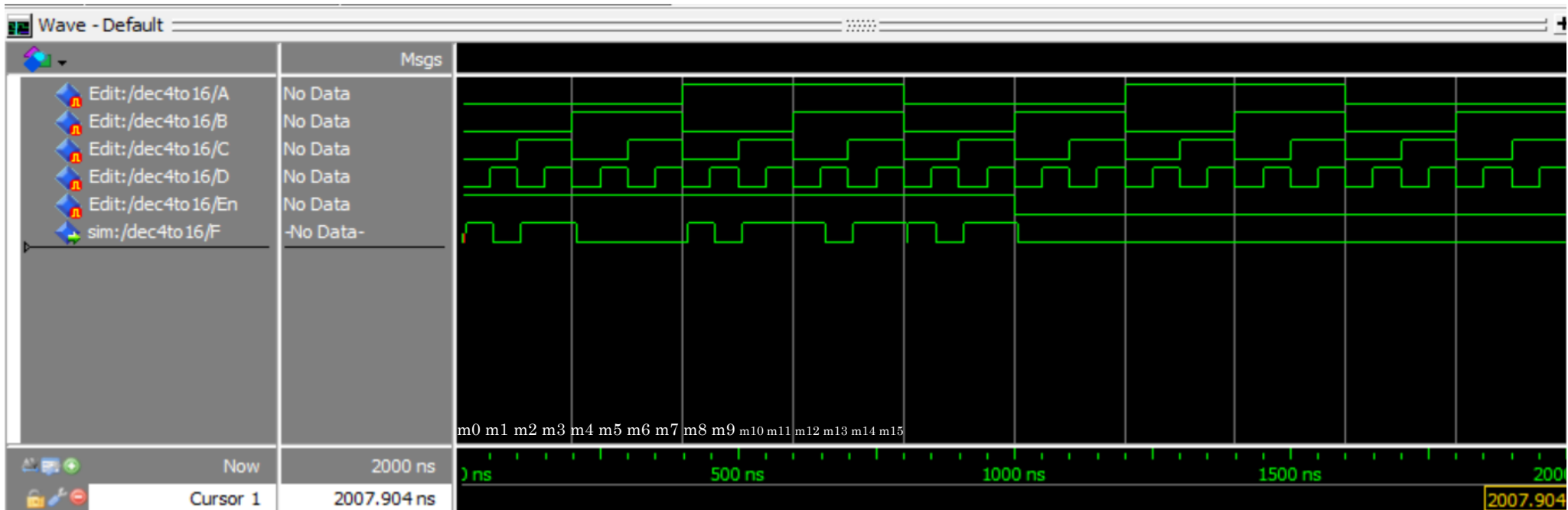
A,B,C,D,Enable이 여러 개의 MUX로 이루어진 Decoder로 구현된 것을 알 수 있다.

4. Circuit Design



식을 Decoder 회로로 구현하면 다음과 같다.

5. Simulation capture



4-to-16 Decoder을 시뮬레이팅 해 보았다. 앞에서 MUX로 구현하였을 때와 마찬가지로 $F(A,B,C,D) = \sum m(0,2,3,8,10,11,12,14,15)$ 의 결과와 같이 F의 값이 0과 1로 결정되었다. 예를 들어 진리표에서 A,B,C,D가 1111 일 때 F가 1이었다. 또다른 예시로 0110 일 때 F는 0이었다. 위의 그림에서 0010이 해당하는 m2의 F값이 1로, 0110에 해당하는 m6의 F값이 결정된 것을 확인할 수 있었다. 또한 Enable이 0일 때에는 F의 값이 무조건 0으로 Output되는 것을 확인할 수 있었다. 이로서 설계한 4-to-16 Decoder가 문제가 없음을 검증하였다.

6. Discussion

저번 실습은 회로를 직접 설계하여 2-level OR logic과 2-level NAND 회로를 구현하고 시뮬레이팅 하는 것이 중점 내용이었다면 이번 실습은 VHDL을 이용하여 코드로 회로를 구현하는 것이 주된 내용이었다. 이번에는 MUX와 Decoder를 이용하여 회로를 구현하였는데 이 과정에서 알 수 있었던 사실들이 있다.

먼저 Decoder은 여러 개의 MUX로 이루어져 있다는 것을 RTL viewer을 통해서 확인할 수 있었다. 그리고 Decoder을 통해 나온 y0부터 y15까지의 값 중 진리표에서 1에 해당되는 y의 값을 or시켜야 진리표와 동일한 결과를 얻을 수 있음을 확인하였다.

이번 실습을 통해서 MUX와 Decoder의 활용법에 대해서 알아보았는데 솔직히 AND gate 등에 비하면 불편한 회로라고 느꼈다. 하지만 MUX와 Decoder만이 가진 경제성이 있어 이러한 회로도 분명 유용하게 쓰일 것이라고 확신한다.

이제 점점 Quartus 프로그램의 사용이 익숙해지고 좀 정이 들어 가는 것 같다. 직접 회로를 구현하는 방법과 VHDL 코드로 구현하는 방법 두가지를 배울 수 있었으며 둘 중 적합한 방식으로 회로를 검증할 수 있게 되었다. 이제 프로그래밍을 할 때 알고리즘 설계 시 내가 사용할 수 있는 적절한 구현 방법의 선택과 회로의 선택 폭이 넓어 진 것 같다.