

1. 다수 함수는 입력에 1이 0보다 많으면 출력이 1이고, 그 이외의 경우에는 출력이 0이다. 문제 1에서와 같은 방법으로 세 개의 입력을 갖는 다수 함수를 설계하되, Specification 단계부터 Technology mapping 단계까지 보이시오.
2. 4-bit equality comparator를 설계하려고 한다. 다음에 제시된 요령에 따라 계층적 설계 방법을 보이시오.
 - (a) 2-bit equality comparator를 설계하시오. 슬라이드 3-5의 block MX를 reuse할 수 있다.
 - (b) (a)에서의 2-bit equality comparator를 reuse하여 4-bit equality comparator를 설계하시오.
 - (c) 슬라이드 3-6에서의 오른쪽 방법으로 계층 구조를 보이시오.
3. Enable 신호를 갖는 2-to-4 디코더를 활용하여 3-to-8 디코더를 설계하시오. 단, 2-to-4 디코더 외의 논리소자의 사용을 최소화하시오.
4. “특별한” 3-to-6 라인 디코더를 설계하려고 한다. 입력으로 000부터 101까지 6개의 조합이 주어진다. 입력 코드들에 대하여 대응되는 출력만 1이 되고 나머지는 0이 되어야 한다. 한 개의 2-to-4 라인 디코더와 다른 한 개의 1-to-2 라인 디코더, 그리고 6개의 AND 게이트를 이용하여 3-to-6 라인 디코더를 설계하시오.
5. 8-to-1 multiplexer를 설계하려고 한다.
 - (a) 다수 개의 2-to-1 multiplexer 만을 사용할 수 있다. 논리도를 보이시오.
 - (b) 다수 개의 4-to-1 multiplexer 만을 사용할 수 있다. 논리도를 보이시오.
 - (c) 두 개의 4-to-1 multiplexer와 한 개의 인버터, 한 개의 OR 게이트를 사용할 수 있다. 단, multiplexer는 enable 신호를 포함한다. 논리도를 보이시오.
6. Multiplexor를 활용하여 Lecture-Hall Lighting Control을 다시 설계하려고 한다. 다음 물음에 답하시오.
 - (a) 4-to-1 multiplexor를 사용하면 슬라이드 3-19의 논리회로를 단순화할 수 있다. 논리도를 보이시오.
 - (b) I_0, I_1, I_2 는 M_0, M_1, M_2 의 함수이다. M_0, M_1, M_2 로 I_0, I_1, I_2 를 생성하기 위한 논리회로를 보이시오.
 - (c) (a)와 (b)를 조합하여 슬라이드 3-19를 대신할 수 있는 논리회로를 보이시오.
7. 2-to-1 Line Quad Multiplexer를 설계하려고 한다. 슬라이드 3-43을 참조하여 물음에 답하시오.
 - (a) 슬라이드 3-43과 유사한 방법으로 2-to-4 Line Decoder와 4×2 AND-OR를 활용하여 설계하시오.
 - (b) (a)를 참조하되, AND, OR, NOT gate만 사용하여 설계하시오.

8. 슬라이드 3-47의 논리회로를 4-to-1 Line Multiplexor로 다시 설계하려고 한다.
- (a) 4-to-1 Line Multiplexor의 입력으로 A와 B를 선택하였다. 입력은 A와 B, 출력은 F인 진리표를 보이시오. 이때 출력은 C와 D의 함수가 된다.
 - (b) (a)에서의 진리표를 활용하여 주어진 기능에 맞도록 설계하시오.
9. 한 개의 8-to-1 라인 멀티플렉서와 한 개의 인버터를 사용하여 다음 부울 함수를 구현하시오.
- $$F(A, B, C, D) = \sum m(2, 4, 6, 9, 10, 11, 15)$$
10. 디코더와 OR 게이트를 사용하여 앞의 문제에서 주어진 부울 함수를 구현하려고 한다.
- (a) 디코더의 크기를 결정하시오. 단, 최소 크기의 디코더를 사용하기로 한다.
 - (b) Logic diagram을 보이시오.