4주차 예비보고서

전공: 컴퓨터공학과 학년: 2학년 학번: 20211522 이름: 김정환

**1.**

.................

텍스트, 도표, 그림, 평면도이(가) 표시된 사진

자동 생성된 설명

................

**2.**

.......................

NAND Gate는 입력이 모두 1일 때 결과로 0을 출력한다. 이외에 두 입력 중 하나라도 0이 있는 경우에는 1을 출력한다. 논리식으로는 ~(AB)로 나타낼 수 있다. NOR Gate의 경우에는 입력이 모두 0일 때만 1을 출력한다. 그 외에는 모두 0을 출력한다. 논리식으로는 ~(A+B)로 나타낼 수 있다. XOR Gate의 경우에는 두 입력의 값이 다른 경우에 1을 출력하고, 같은 경우에는 0을 출력한다. 논리식으로는 (~A)B + A(~B)로 나타낼 수 있다.

도표, 라인, 디자인이(가) 표시된 사진

자동 생성된 설명

NAND의 기호

스케치, 도표, 라인, 디자인이(가) 표시된 사진

자동 생성된 설명

NOR의 기호

스케치, 도표, 라인, 그림이(가) 표시된 사진

자동 생성된 설명

XOR의 기호

.........................

**3.**

.......................

NAND Gate는 AND Gate에 Inverter를 직렬 연결한 형태의 모습이라고 생각할 수 있다. AND 연산 이후에 NOT 연산이 진행되는 방식으로 AND Gate의 모든 출력에 대해 NOT 처리를 해준 방식이다. 마찬가지로 NOR Gate 역시도 OR Gate에 Inverter를 직렬 연결한 방식으로 OR Gate의 모든 출력에 대해 NOT 연산을 진행한다.

그러나 transistor-level에서의 회로의 모습을 그린 것을 3주차의 1번 문항과 이번 주차의 1번 문항과 비교해보면, 회로 자체의 구현에 있어서는 NAND Gate와 NOR Gate를 먼저 구현하고 해당 회로에 Inverter를 직렬 연결하는 방식으로 AND와 OR Gate가 구현되었다고 볼 수 있다.

.........................

**4.**

.......................

AND-OR-INVERT logic은 AND Gate와 OR, INVERTER의 조합으로 구성된 조합 논리라고 할 수 있다. 이는 AOI Gate라고도 부르며, 입력 값의 개수에 따라서 앞에 붙는 숫자가 달라진다. 2-1 AOI Gate는 먼저 두 입력에 대해 AND 연산을 수행한 후에 해당 출력과 남은 입력 하나와 OR과 NOT 연산을 수행한다. 다음으로 2-2 AOI Gate는 각각 두 입력씩 AND 연산을 수행한 후, 두 출력에 대해서 OR과 NOT 연산을 수행한다. AOI Gate의 방식으로 회로를 구성하는 것은 AND, OR, NOT을 각각 구현한 경우보다 총 트랜지스터의 수가 적다는 점에 장점이 있다. 이는 전력 사용의 감소나 제조 비용이 낮아지는 것을 의미한다.

.........................

**5.**

.......................

XOR 연산을 논리식으로 나타낼 때 우선 나타낼 수 있는 방법은 A(~B)+(~A)B로 나타낼 수 있다. 이 경우에는 AND, OR, NOT을 이용해서 XOR을 나타내는 방식으로 구현했다고 할 수 있다. 다음으로는 NAND Gate를 이용하여 XOR Gate를 구현하는 방법도 있다. 이는 위에서 AND, OR, NOT을 이용해서 구현한 식을 변형하여 만들게 되는데 각각의 연산규칙에 의해 정리하면, NAND(NAND(~A,B), NAND(A, ~B)))의 방식으로 정리할 수 있다. 이 방식에서는 NAND(입력값1, 입력값2)의 방식으로 나타냈다. NOR로도 XOR을 구현할 수 있는데, 이는 NAND로 만든 것처럼 이를 NOR로 바꿔주면 XNOR이 나오는 것을 이용하면 된다. NOR로 바꿔준 후 마지막에 NOT을 붙여 구현하면 XNOR에 NOT 연산을 추가하여 XOR이 나오게 된다.

.........................

**6.**

.......................

회로를 구현할 때 게이트의 수나 입력 개수에 따라서 구현이 결정되는데, 완전한 최소식을 얻는 과정에 필요한 것으로는 카르노맵이 있다. 카르노맵은 우선 2변수, 즉 입력이 2개일 경우에는 2X2 행렬로 나타내며, 진리표의 값을 위치에 맞게 넣어준다. 그 후 인접한 같은 값들 중에서 변수가 다른 부분을 확인하여 어떤 변수가 고정되는지 확인하면 식을 간소화할 수 있게 된다. 3변수일 때는 1개의 변수와 2개의 변수로 나눠서 2X8 행렬의 형태로 나타낸다. 여기서도 0을 기준으로 하던, 1을 기준으로 하던 기준으로 정한 값은 모두 묶일 수 있도록 정리한다. 마지막으로 4변수일 때는 4X4의 행렬의 형태로 변수를 2개씩 묶어서 표로 나타낸다. 이를 이용하면 식만을 보고 정리할 때보다 제일 간소화된 식을 얻는데 있어서 좀 더 유리하고 더 간단히 같은 결과를 나타낼 수 있게 된다.

.........................