4. 부울함수의 간소화 및 구현
  1. 개요 [101]
    1. 대수적 방법(algebraic method)
    2. 도표 방법(map method)
      카르노도표(karnaugh map)를 사용
      부울함수의 각 항을 하나의 곱 형태로 쉽게 간소화
      변수가 많아짐에 따라 도표의 구성이 매우 복잡해지므로 6개 이하의 변수를 가진 부울함수만을 대상으로 함
    3. 테이블 방법(tabular method)
  2. 카르노도표 방법 [102]
    1. 개요
      부울대수 공식을 이용한 방법의 단점
        다음 단계의 처리를 예견할 수 있는 일반적인 방법이 없음
        가장 간소화된 식을 구했는지 판단하기 어려움
      도표에 의한 간소화 방법(map simplification method)
        카르노 도표를 이용함으로써 부울함수 쉽게 간소화
        카르노 도표
          여러 개의 사각형으로 된 다이어그램
          사각형은 각각 하나의 최소항 또는 최대항을 나타낸다.
            어떤 부울함수든지 표준형의 곱의 합(sum of products) 또는 합의 곱(product of sums) 형태로 표현할 수 있으므로 부울함수 내의 최소항(또는 최대항)이 차지하는 도표 내의 면적으로 직관적으로 부울함수를 간소화할 수 있다.
          n 변수 카르노도표
            입력변수의 수에 따라 기본 도표 형태가 달라짐
            2^n개의 사각형으로 구성 (n은 입력변수의 수)
          방법
            부울함수를 정규형(최소항의 합형태와 최대항의 곱형태) 부울함수로 변형
            카르노 도표를 이용하여 표준형(곱의 합형태와 합의 곱형태)로 표시
        1) 최소항의 합형태를 곱의 합형태로 간소화하는 순서
          1. **카르노도표 작성** 입력변수의 수 n에 따라 n변수 카르노도표를 작성한다. 이때 도표는 2^n개의 정사각형으로 구성된다. 2. **최소항 표시** 정규형 부울함수의 최소항 인덱스에 대응되는 칸을 1로 표시한다. 3. **인접 사각형 묶기** 1로 표시된 칸 중 서로 인접한 것끼리 묶는다. 이때 묶음 하나는 크되, 묶음 전체 개수는 최소가 되도록 한다. 4. **각 묶음의 항 결정** 묶음이 변수 X에 대해 - X=1인 곳에만 존재하면 → 곱항에 X를 남긴다. - X=0인 곳에만 존재하면 → 곱항에서 X(및 보수 $\bar{X}$)를 제거한다. - X=0·X=1 양쪽에 걸쳐 있으면 → 곱항에서 X 변수를 완전히 소거한다. 5. **최종식 구성** ④에서 결정한 곱항들을 논리합(OR, '+')으로 연결하여 **간소화된 표준형(곱의 합형태)**을 얻는다.
        2) 최대항의 곱형태를 합의 곱형태로 간소화하는 순서
          1. **카르노도표 작성** (1)과 동일하게 n변수 카르노도표를 작성한다. 2. **최대항 표시** 정규형 부울함수의 최대항 인덱스에 대응되는 칸을 0으로 표시한다. 3. **인접 사각형 묶기** 0으로 표시된 칸 중 서로 인접한 것끼리 묶는다. 4. **각 묶음의 항 결정** 묶음이 변수 X에 대해 - X=1인 곳에만 존재하면 → 합항에 X를 남긴다. - X=0인 곳에만 존재하면 → 합항에 $\bar{X}$를 남긴다. - X=0·X=1 양쪽에 걸쳐 있으면 → 합항에서 X 변수를 완전히 소거한다. 5. **최종식 구성** ④에서 결정한 합항들을 논리곱(AND, '·')으로 연결하여 **간소화된 표준형(합의 곱 형태)**을 얻는다.
        3) 인접 사각형
          (1) 정의
            '두 정사각형이 서로 인접한다'의 의미
              카르노도표에서 각 정사각형은 하나의 최소항을 의미[104]
              부울공식인 $\bar{X} + X = 1$을 이용하여 두 최소항을 간소화
              다른 모든 문자는 서로 동일하되 오직 하나의 문자만 서로 보수관계에 잇을 때 두 정사각형은 서로 인접하다
          (2) 인접 사각형끼리 묶는 법
            첫째, 한 묶음 내의 정사각형 수는 2^n (n=0, 1, 2, …, n)개가 되도록 묶는다.
            둘째, 한 묶음은 크게, 전체 묶음의 수는 작게 묶는다.
              한 묶음 내의 정사각형 수는 2^n (n=0, 1, 2, …, n)개로 묶어야 한다.
                카르노도표를 이용한 간소화 방법은 부울공식 $X+\bar{X}=1$을 이용하기 때문에, 먼저 2개의 정사각형을 묶어 하나의 곱항을 만들고, 이러한 곱항 간에 다시 부울공식 $X+\bar{X}=1$을 사용한다면 역시 2개의 곱항을 묶을 수 있는데, 이것은 모두 2^2=4개의 정사각형을 묶는 결과이다.

- 특징
  - 한 묶음을 크게 묶으면 곱항 내의 문자의 수가 더 많이 줄어들게 되고, 전체 묶음의 수를 작게 묶으면 곱항의 수가 작게 된다.
  - 곱항 내의 문자의 수가 줄어들면 **AND** 연산의 수 또는 **AND** 연산의 입력수가 줄어들고, 곱항의 수가 줄어들면 **OR** 연산의 수 또는 **OR** 연산의 입력수가 줄어드는 결과가 된다.
  - 카르노 도표 : 부울함수를 정규형으로 표시할 수 있는 모든 표현식을 시각적으로 표현할 수 있는 방법
  - 가장 간소화된 대수식
    - 최소수의 항으로 구성되고, 각 항에서 문자의 수를 가능한 한 적게 한 것으로, 게이트 수를 최소화 시키고 게이트 입력수를 최소화시킨 논리회로도로 구현된다.
    - (때때로) 간소화 기준을 만족하는 둘 이상의 부울식을 찾을 수도 있음. 해가 여러 개 있는 것임

- 2. 카르노도표 방법(2) [106]
  - 2. 2변수 카르노도표
    - 2개의 변수를 가지는 부울함수에는 4개의 최소항이 있음.
    - 카르노도표 나타내기
    - 카르노도표 간소화
      - 인접 사각형 묶기
        - 묶음 2
          - 변수 X = 1, 변수 Y는 0이거나 1인 경우
            - 곱항 X로 간소화
        - 묶음 3
          - X는 0이거나 1 Y = 1
            - 곱항 Y로 간소화
      - 대수적 처리의 증명
  - 3. 3변수 카르노도표
    - 카르노도표 나타내기
      - Y와 Z 값이 00, 01, 10, 11과 같이 순차적 2진 계수가 아니라 인접한 열에서 다음 열의 값이 1비트 차이가 생기도록 한 것에 유의
        - 도표상에서 시각적으로 인접한 두 정사각형이 실제 정의에 따라서도 인접한 것이 되도록 하기 위함
          - $\bar{X} + X = 1$로 간소화하기 때문에 인접시 생략하려면 단 하나만 차이가 있어야한다. => 그레이 코드로 표기
      - 시각적으로는 서로 붙어 있지 않지만 정의에 의해서는 서로 인접한 정사각형이 있음
        - [그림 4.4(a)에스 m_0와 m_2는 서로 떨어져 있으나 변수 Y만 서로 보수관계에 있고 나머지 변수는 모두 $\bar{X}\bar{Z}$로 동일하므로 서로 인접한 정사각형에 해당
    - 예제 4.1
    - 예제 4.2
      - 한 묶음은 크게, 묶음의 수는 작게 묶는다.
        - [그림 4.7(c)]를 살펴보자. 이 경우는 [그림 4.7(b)]의 묶음 1을 묶음 3과 묶음 4의 2개로 나누어 묶는다. 묶음 3과 묶음 4는 각각 **$\bar{X}$ Y**와 **X Y**로 간소화되는데, 이들은 $\bar{X}Y + XY = (\bar{X} + X)Y = Y$로 다시 간소화되고 이것은 묶음 1을 간소화한 것과 동일한 결과이다.
        - 또한 묶음 5는 1개의 최소항으로서 **$\bar{X}\bar{Y}Z$**이다. 그러나 묶음 2처럼 묶는다면 **$\bar{X}\bar{Z}$**로 1개의 문자를 줄일 수 있다.
          - 그림 (c)처럼 묶는다면 간소화된 부울함수는 $P(X,Y,Z) = \bar{X}Y + XY + \bar{X}\bar{Y}Z$ 가 되어 더 복잡한 부울함수를 얻을 수 있다.
            - 한 묶음을 작게 묶고, 묶음의 수도 (b)의 2개보다 많은 3개로 하여 부울함수가 충분히 간소화되지 못하는 결과를 초래한다.
    - 예제 4.3.
  - 4. 4변수 카르노도표 [113]
    - 특징
      - 16개의 정사각형으로 구성됨
      - 4개의 2진 변수 W, X, Y, Z로 구성
      - 각각의 변수는 8개의 정사각형에서는 보수를 취하지 않고, 다른 8개의 정사각형에서만 보수를 취한다.
      - m_8과 m_{10} 그리고 m_1과 m_6은 각각 서로 인접한 정사각형이다.
    - 간소화

① 하나의 정사각형은 4개의 문자의 곱항(최소항)으로 표시된다. ② 2개의 인접한 정사각형은 3개의 문자의 곱항으로 표시된다. ③ 4개의 인접한 정사각형은 2개의 문자의 곱항으로 표시된다. ④ 8개의 인접한 정사각형은 1개의 문자의 항으로 표시된다. ⑤ 16개의 인접한 정사각형은 전체 도표를 포함하므로, 함수값이 상수 1인 부울함수로 간소화된다.

- 예제 4.4
  - 인접 사각형끼리 묶는다. 묶음 1은 W=0 또는 1, X=1, Y=0, Z=0 또는 1인 위치에 있으므로 $X\bar{Y}$로 간소화된다. 또한 묶음 2는 W=0, X=0, Y=0 또는 1, Z=1인 위치에 있으므로 $\bar{W}\bar{X}Z$로 간소화된다. 끝으로 묶음 3은 W=0 또는 1, X=0, Y=1, Z=1인 위치에 있으므로 $\bar{X}YZ$로 간소화된다. 따라서 간소화된 부울함수는 다음과 같이 3개의 곱항을 논리합으로 연결하여 얻는다. $F = X\bar{Y} + \bar{W}\bar{X}Z + \bar{X}YZ$
- 예제 4.5
  - 간소화된 식
    - $F = C + B\bar{D} + \bar{A}BD$
  - 논리회로도
- 예제 4.6
  - 간소화된 부울함수

2. 카르노도표 방법(3) [120]

5. 5변수 카르노도표

- 5개 변수 이상의 카르노도표는 사각형 수가 많아서 인접 사각형을 찾는 것이 매우 어렵기 때문에 사용하기 쉽지 않다.
- 특징
  - 2비트의 2진 그레이 코드와 3비트의 2진 그레이 코드의 개념에서 인접 사각형의 정의가 다르다
  - 2개의 2진 그레이 코드는 00, 01, 11, 10의 순서가 되며, 서로 이웃하는 사각형, 맨 왼쪽과 맨 오른쪽, 맨 위와 맨 아래가 서로 인접관계가 성립되지만, 3비트의 2진 그레이 코드는 000, 001, 011, 010, 110, 111, 101, 100의 순서이므로, 서로 이웃하는 사각형뿐만 아니라 이웃하지 않는 사각형도 인접관계가 성립될 수 있다.
- 예제 4.7
  - 간소화 결과
  - 간소화 정리
    - 5변수 카르노맵에서 크기 2^k 묶음은 5−k개의 리터럴을 갖는 곱항을 생성하며, 전체 32칸을 모두 묶으면 상수 1이 됩니다.
      - ① 1개의 정사각형 → 5개의 문자의 최소항 ② 2개의 정사각형 → 4개의 문자의 곱항 ③ 4개의 정사각형 → 3개의 문자의 곱항 ④ 8개의 정사각형 → 2개의 문자의 곱항 ⑤ 16개의 정사각형 → 1개의 문자의 곱항 ⑥ 32개의 정사각형 → 부울함수 ≡ 1

6. 6변수 카르노도표

- 6개의 변수를 3개씩 나누어 행과 열에 배치하고, 행과 열에는 각각 8개의 변수에 대한 8개의 최소항이 그레이 코드의 순서로 배열된다
- 곱항 내의 문자 수

7. 무관 조건(don't care condition)

- 모든 변수의 조합은 언제나 1 또는 0의 함수값을 가지며, 이 함수값은 일정한 입력변수의 조합에서는 언제나 일정한 출력값을 가진다.
  - 디지털 논리회로에서 입출력상태를 나타내는 진리표는 입력변수의 조합에 따라 함수값이 0 또는 1을 가진다. 함수값이 1인 입력변수를 AND 연산의 항으로 표시한 것이 최소항이고, 이들 최소항을 OR 연산으로 결합한 것이 최소항의 합 형태인 정규형 부울함수이다.
- 어떤 논리회로에서는 입력변수의 조합에 따라서함수값이 발생하지 않는 경우 또는 0이나 1중에서 어떠한 함수값이 출력값으로 나와도 무관한 경우가 있다.
  - 즉, 입력변수의 조합에 따라 출력값이 1 또는 0의 일정한 값이 되지 않는 경우가 있다.
    - BCD 코드에서는 10진수의 한 자리를 2진수의 네 자리로 하여 모든 10진 숫자를 2진수로 표현한다. 그러나 2진수의 4비트는 16개의 비트 조합으로 구성되기 때문에 그중에서 10개만 숫자 표현에 사용하고, 나머지 6개의 비트 조합(1010, 1011, 1100, 1101, 1110, 1111)은 사용되지 않는다.
    - 따라서 BCD 코드를 사용하는 디지털 논리회로에서는 6개의 사용되지 않는 비트 조합은 발생하지 않는다는 가정하에 동작한다. 6개의 사용되지 않는 비트 조합에 대해서는 무관하게 논리회로를 구성하기 때문에 이들 입력 변수의 조합들에 대한 함수의 출력값은 어떠한 값을 가지더라도 무관하다.
- 무관조건의 표현
  - 표현 예시
    - $d(A, B, C) = \Sigma m(0, 1, 7)$
      - 3변수 부울함수에 대한 무관조건으로 3개의 변수 A, B, C를 2진수 열로 묶어 표시할 때 각각 000, 001, 111일 때는 해당 부울함수의 출력이 0이든 1이든 무관하다는 것을 의미한다.
  - 부울함수의 간소화에서 인접 사각형의 조합을 선택할 때, 무관조건인 X는 더욱 간소화된 부울함수를 얻을 수 있도록 큰 항으로 묶을 때 1 또는 0으로 사용될 수 있으며, 간소화 과정에서 큰 항의 묶음에 포함되지 않는 무관조건은 사용되지 않는다.

- 예제 4.8
  - 다음 부울함수를 간소화하시오
    - (1) F(W,X,Y,Z) = Σm(0,3,6,9), 단 d(W,X,Y,Z) = Σm(10,11,12,13,14,15)과 같다.
      - 풀이
        - ① 주어진 부울함수로부터 카르노도표에 해당하는 사각형에 각각 함숫값을 표시한다. ② 인접관계를 이용하여 사각형의 함숫값이 1이거나 ×(곱의 합 형태로 간소화하는 경우) 또는 0이거나 ×(합의 곱 형태로 간소화하는 경우)인 경우 큰 항으로 묶는다.
          - a. 곱의 합 형태로 간소화하는 경우[126]
          - b. 합의 곱 형태로 간소화하는 경우 [126]
    - (2) F(W,X,Y,Z) = Σm(1,3,7,15), 단 d(W,X,Y,Z) = Σm(0,2,9,11,13)과 같다.
      - 풀이
        - ① 주어진 부울함수로부터 카르노도표에 해당하는 사각형에 각각 함숫값을 표시한다. ② 인접관계를 이용하여 사각형의 함숫값이 1이거나 ×(또는 0이거나 ×)인 경우 큰 항으로 묶는다.
          - a. 곱의 합 형태로 간소화하는 경우[128]
          - b. 합의 곱 형태로 간소화하는 경우 [128]
    - 예제 4.8에서는 무관조건을 사용하여 곱의 합형태와 항븨 곱형태인 간소화된 부울함수를 구하기 때문에 무관조건이 없이 간소화된 부울함수와는 다르다.
- 8. 기타 카르노도표
  - XOR와 XNOR 논리게이트에 대한 카르노도표
  - 기호 ⊕(XOR)는 다음과 같이 등가의 논리식으로 바꾸어 일반의 논리함수로 변환할 수 있다.
    - 특히 3변수의 경우 다음과 같은 논리식으로 바꿀 수 있다.
    - 다중변수 XOR 연산은 홀수함수(odd function)로 정의된다.
      - 위의 식은 단지 한 변수가 1이거나 또는 세 변수 모두가 1인 경우에만 3변수 XOR 함수가 1이 된다는 것을 보여 주고 있다. 그러나 2변수 함수에서는 오직 한 변수만 1이 되어야 한다.
      - 3변수 또는 그 이상의 변수를 쓰면 홀수 개의 변수가 1이어야 한다.
  - 일반적으로 n개의 변수를 XOR로 표현한 부울함수는 $2^n/2$개의 최소항을 갖는다.
    - 이때 각 최소항에 대응하는 등가의 2진수는 홀수 개의 1을 갖는다. 따라서 홀수함수에 대한 정의는 도표로 쉽게 확인할 수 있다.
  - 3변수 XOR 연산으로부터 도출한 함수는 그 2진수값이 001, 010, 100, 111인 4개의 최소항의 논리합으로 표현할 수 있다. 이들 2진수 각각은 1을 홀수 개 가지고 있다.
  - 함수에 포함되지 않은 나머지 4개의 최소항은 000, 011, 101, 110으로, 이들은 짝수 개의 1을 가지고 있다.
  - 그림
    - 해석
      - 도표에서 보면 1로 표시된 8개의 최소항이 홀수함수를 만들고 있음을 알 수 있다.
      - 그러나 1로 표시하지 않은 최소항들은 1이 짝수 개 있으며 홀수함수의 보수를 만들고 있다는 것도 알 수 있다.
      - 홀수함수는 [그림 4.24]와 같이 XOR 게이트를 이용하여 구현할 수 있다.
      - XNOR 논리게이트는 XOR의 부정이기 때문에 XOR의 카르노도표인 [그림 4.23]의 0과 1을 뒤바꾸어 그리며, XNOR의 논리도는 [그림 4.24]의 출력에 NOT 게이트를 추가함으로써, 또는 출력 게이트인 XOR 게이트에서 XNOR 게이트로 대체함으로써 구할 수 있다.
- 3. NAND 게이트와 NOR 게이트를 이용한 구현방법 [132]
  - 1. 개요
    - 모든 부울함수는 AND, OR, NOT 게이트를 사용하여 간단하게 구현할 수 있다.
      - 실제 회로를 설계할 때, AND와 OR 게이트보다 NANA와 NOR 게이트를 사용하여 부울함수를 구현하는 경우가 더 많다.
      - 전자회로로 제작하기가 쉽고, 이들을 사용하여 AND, OR, NOT 논리연산을 모두 구현할 수 있기 때문이다.
      - 디지털 논리회로 설계에서는 AND, OR, NOT 논리연산을 사용하여 구현된 부울함수를 NAND와 NOR 게이트의 논리로 변환하여 구현한다.
    - 부울함수는 최소항의 합이나 최대항의 곱으로 표현되는 정규형으로 나타낼 수 있다.
      - 최소항의 합형태는 간소화 과정을 거쳐서 곱합의 합으로 표현될 수 있고, 이러한 곱항의 합은 NAND 게이트만을 이용하여 부울함수를 구현할 수 있다.
      - 최대항의 곱형태는 간소화 과정을 거쳐서 합항의 곱으로 표현될 수 있고, 합항의 곱은 NOR 게이트만을 이용하여 부울함수를 구현할 수 있다.
  - 2. NAND 게이트를 이용한 논리회로 구현방법
    - 1) NAND 게이트 개념 및 구성

NAND 게이트는 논리적 곱의 보수(AND-NOT)를 수행하는 기능을 하며, 2개 이상의 입력과 1개의 출력으로 구성된다.
NAND 게이트의 수행 방식은 - 입력 중 1개 이상의 입력이 논리-0이면 출력은 논리-1이고 - 모든 입력이 논리-1이면 출력은 논리-0이 된다.

NAND 게이트만으로 [그림 4.25]와 같이 AND, OR, NOT의 논리연산을 구성할 수 있다.

표현

- AND-NOT의 NAND 게이트는 출력부에 작은 원을 붙인 AND 기호로 나타낸다. - NOT-OR의 NAND 게이트는 모든 입력부에 작은 원을 붙인 OR 기호로 나타낸다.

AND-NOT의 형태와 NOT-OR의 형태는 드모르간의 법칙에 의해 NAND 게이트로도 구현할 수 있다는 것을 증명 가능하다.

2) 2단계 구현

임의의 부울함수를 NAND 게이트만을 이용하여 2단계로 논리회로도를 구현하는 과정

① 간소화하여 곱의 합 형태로 나타낸다.

② 2개 이상의 입력의 곱항을 NAND 게이트로 나타내고, 하나의 입력의 곱항을 NOT 게이트로 나타낸다.

③ ②에서의 출력을 AND-NOT 혹은 NOT-OR 형태를 사용하여 2단계 게이트로 그린다.

예시

부울함수 F = XYZ + WX를 NAND 게이트로 구현

①번 과정을 수행하면 [그림 4.27(a)]의 AND-OR 논리회로도가 된다. 그다음에 ②번과 ③번의 과정을 수행하면 NOT-OR 기호를 사용한 [그림 4.27(b)]의 논리회로도가 되거나, AND-NOT 기호를 사용한 [그림 4.27(c)]의 논리회로도가 된다.

예제 4.9

풀이

부울함수 F는 곱의 합 형태로 되어 있기 때문에 ①번 과정을 수행하면 [그림 4.28(a)]의 AND-OR 논리회로도가 된다. 그다음에 ②번과 ③번의 과정을 수행하면 NOT-OR 기호를 사용한 [그림 4.28(b)]의 논리회로도가 되거나, AND-NOT 기호를 사용한 [그림 4.28(c)]의 논리회로도가 된다.

3) 다단계 구현

부울함수를 표준형으로 고쳐서 NAND 게이트를 사용하여 논리회로를 구현하면 2단계로 논리회로도를 구성하게 되지만, 그렇지 않은 경우에는 3단계 이상, 다단계의 논리회로도로 구현된다.

다단계 논리회로도를 설계하려면 먼저 해당 부울함수를 AND, OR, NOT 게이트로 표현한다. 다음으로 그 논리회로도에 있는 모든 게이트를 모두 NAND 게이트로 변환한다.

1) 모든 AND 게이트를 AND-NOT 기호의 NAND 게이트로 변환한다.

2) 모든 OR 게이트를 NOT-OR 기호의 NAND 게이트로 변환한다.

3) 논리회로도에서 모든 작은 원을 점검하여, • 동일한 선상의 두 원은 서로 상쇄시키고, • 상쇄되지 않은 원은 NOT 게이트를 추가하거나 입력변수에 보수를 취한다.

그림 4.29

AND 게이트를 AND-NOT 기호로 바꾸고, OR 게이트를 NOT-OR 기호로 바꾼다. 따라서 부울함수 F를 NAND 게이트만으로 논리회로도를 구성할 수 있다.

[그림 4.29(a)]에서의 입력 $\bar{X}$, Y, W, $\bar{W}$가 (b)로 변환되는 과정에서 각각 X,$\bar{Y}$, $\bar{W}$, W로 바꾸게 된다.

이들을 외부 게이트로 보수가 취해져야만 하기 때문이다.

예제 4.10

AND 게이트를 AND-NOT 게이트로 변환한 후 출력 F가 동일선상에 상쇄되는 작은 원이 없기 때문에 하나의 NOT 게이트를 추가해야 한다.

3. NAND 게이트와 NOR 게이트를 이용한 구현방법 (2) [139]

3. NOR 게이트를 이용한 논리회로 구현방법

1) NOR 게이트 개념 및 구성

NOR 게이트는 논리적 합의 보수(OR-NOT)를 수행하는 기능이 있고, 2개 이상의 입력과 1개의 출력으로 구성된다.

NOR 게이트의 수행 방식으로는 입력 중 1개의 입력만이라도 논리-1인 경우 출력은 논리-0이고, 그렇지 않은 경우에는 논리-1이 된다.

NOR 게이트의 논리연산

NAND 게이트와 마찬가지로 [그림 4.31]과 같이 AND, OR, NOT 게이트 등을 구성할 수 있기 때문에 논리회로를 설계할 때 매우 유용

[그림 4.32]와 같은 기호를 사용한다. - OR-NOT의 NOR 게이트는 출력부에 작은 원을 붙인 OR 기호로 나타낸다. - NOT-AND의 NOR 게이트는 모든 입력부에 작은 원을 붙인 AND 기호로 나타낸다.

OR-NOT 형태와 NOT-AND 형태는 드모르간 법칙에 의해 NOR 게이트로 나타낼 수 있다는 것을 증명할 수 있다.

2) 2단계 구현

임의의 부울함수를 NOR 게이트를 사용하여 2단계로 논리회로도를 구현하는 과정

① 간소화하여 합의 곱 형태로 나타낸다.

② 2개 이상 입력의 합항을 NOR 게이트로 나타내고, 하나의 입력의 합항을 NOT 게이트로 나타낸다.

③ ②에서의 출력을 OR-NOT 혹은 NOT-AND 형태를 사용하여 2단계 게이트를 그린다.

그림 4.33

위의 과정을 사용하여 부울함수 F = W(X + Y)를 구현해 보자. 함수 F는 합항의 곱 형태로 되어 있기 때문에 ①번 과정을 수행하면 [그림 4.33(a)]의 AND–OR 논리회로도가 된다. 그다음에 ②번과 ③번의 과정을 수행하면 [그림 4.33(b)]의 논리회로도가 된다. 그리고 함수 F에서는 입력 W에서 NOT을 제거하고 $\bar{W}$로 표시된다. NOR 게이트를 사용한 2단계 구현과정을 모두 수행한 결과는 [그림 4.33(c)]에서의 논리회로도가 된다.

3) 다단계 구현

절차

① 모든 OR 게이트를 OR-NOT 기호의 NOR 게이트로 변환한다.

② 모든 AND 게이트를 NOT-AND 기호의 NOR 게이트로 변환한다.

③ 논리회로도에서 모든 작은 원을 점검하여, – 동일한 선상의 두 원은 서로 상쇄시키고, – 상쇄되지 않은 원은 NOT 게이트를 추가하거나 입력변수에 보수를 취한다.

그림 4.34