3. 논리게이트와 부울대수

- **1. 논리연산과 논리 게이트 [63]**
  - **1. 논리 연산**
    - **1) 개요**
      - 수학적 함수
      - 그래프
      - 전자회로
        - 입력값과 출력값이 이산값인 경우(그래프로 입력전압과 출력전압을 표시할 필요가 없다.) 입출력관계를(d)와 같이 진리표(truth table)로 표현하는 것이 보다 효율적이다
    - **2) 함수 표현**
      - 함수와 관련된 용어
        - 상수
          - 수를 모아 놓은 수집합에서 변하지 않는 값
        - 변수
          - 변할 수 있는 값을 나타내는 기호
          - 유형
            - 독립변수
            - 종속변수
        - 식
          - 상수와 변수를 연산자로 연결한 것
        - 함수
          - 입력에 따라 변수가 어떻게 변하는지를 표현하기 위해 '변수=식' 형태로 나타낸 것
        - 논리집합
          - {0, 1}을 부울(Boolean) 집합이라고도 부름
          - 논리상수, 논리변수, 논리연산, 논리함수 등 용어 사용 가능
    - **3) 논리집합과 논리연산**
      - 논리연산
        - 2개의 이산값에 적용되는, 논리적 의미가 있는 연산
        - 1과 0으로 표현
        - 논리변수
      - 대표적 논리 연산의 유형
        - AND
          - X·Y=F 또는 XY=F는 “X AND Y는 F와 같다”
        - OR
          - X+Y=F는 “X OR Y는 F와 같다”
        - NOT
          - $\bar{X}$=F는 “X의 NOT은 F와 같다
      - 논리변수는 1 아니면 0으로 오직 2개의 값만 갖는다.
  - **2. 논리 게이트**
    - 개요
      - 논리회로(logic circuit)
        - 적절하게 입력된 신호를 가지고 논리적 연산을 수행하여 출력신호를 생성하는 전자적 회로
    - 1) 기본 논리게이트
      - AND 게이트
        - 논리적 곱 수행
        - 2개 이상이ㅡ 입력과 1개의 출력으로 구성
      - OR 게이트
        - 논리적 합 수행
        - 2개 이상의 입력과 1개의 출력으로 구성
      - NOT 게이트
        - 인버터(inverter) 기능 수행

- 2) NAND 게이트와 NOR 게이트
  - NAND 게이트
    - 논리적 곱의 보수(AND-NOT)
  - NOR 게이트
    - 논리적 합의 보수(OR-NOT)
- 3) XOR 게이트와 XNOR 게이트
  - XOR(Exclusive-OR) 게이트
    - 입력되는 2비트에 대해 서로 같은지 틀린지를 비교
    - 패리티 검사(parity check), 그레이 코드(gray code) 변환기, 가산기(adder), 감산기(subtracter) 등의 조합논리회로에 많이 사용됨
    - $X \oplus Y = X\bar{Y} + \bar{X}Y$
  - XNOR(Exclusive-NOR) 게이트
    - XOR 연산의 결과에 NOT 연산
    - 2진 비교기(binary comparator)
    - $X \oplus Y = \bar{X}Y + X\bar{Y}$
- 4) 주요 논리게이트 정리 [73]

2. 부울대수 [74]
- 1. 개요
  - 부울대수(Boolean algebra)
    - 0 또는 1의 값을 갖는 논리변수와 논리연산을 다루는 대수
      - 논리연산
        - AND, OR, 보수
  - 부울함수(Boolean function)
    - 논리변수의 상호관계를 나타내기 위해 부울변수, 부울상수(0 또는 1), 부울연산기호(·, +, ¯), 괄호 및 등호 등으로 나타내는 대수적 표현
    - $F = X \cdot \bar{Y} + X \cdot Y \cdot Z + \bar{X} \cdot Y \cdot Z$ (식 3.1)
      - 이 부울함수의 값(출력 F)은 입력변수 X, Y, Z가 가질 수 있는 모든 값을(논리변수로 0 아니면 1) 각각 (식 3.1)의 우변 식에 대입함으로써 얻을 수 있다.
    - 진리표
      - 논리연산의 결과와 결합한 0과 1의 모든 조합을 표로 나타낸 것이다.
    - 주어진 부울함수를 논리회로도(logic circuit diagram)로 구현한다
      - 논리게이트로 구성되는 회로도를 작성하는 것
      - 독립변수
        - 함수를 구성하는 독립변수들(X, Y, Z)은 회로의 입력
      - 종속변수 F
        - 역시 논리변수로 회로의 출력
  - 주어진 부울함수에 대한 지리표는 오직 하나이지만 동일한 진리표를 만족하는 부울함수는 여러 개가 될 수 있다.
    - 진리표와 부울함수의 1:n 관계와 부울함수와 논리회로도의 1:1 관계
    - 부울함수를 구성하는 부울연산과 부울변수의 수는 결국 게이트의 수 그리고 각 게이트의 입력수와 직접적으로 관계가 있으므로 이들의 수를 줄이는 만큼 논리회로는 단순해진다.   따라서 주어진 부울함수를 가장 단순화된 논리회로로 구현하려면 부울함수를 가능한 한 단순한 식으로 변환해야 한다.
- 2. 기본공식
  - 기본공식
    - 공식 (15) - 분배법칙
    - 드모르간(De Morgan) 법칙- 공식 (16), (17)
      - (16)은 변수 전체 합의 보수는 각 변수의 보수의 곱과 같다
      - (17)은 변수의 전체 곱의 보수는 각 변수의 보수의 합과 같다
    - 흡수정리(absorption theorem) - 공식 (18)
  - 쌍대성 원리 (principle of duality)
    - 부울대수식에서 논리연산자인 +와 ·, 그리고 논리상수 1과 0을 맞바꾸면 상대형태(dual form)를 얻을 수 있다.
- 3. 부울함수의 대수적 간소화
  - 부울함수의 대수적 간소화(algebraic simplification)

- 주어진 부울함수에 대하여 부울대수의 공식을 이용하여 변환한 다음, 변환된 여러 함수 중에서 가장 간단한 형태의 함수를 찾아내는 것
- 부울함수는 AND, OR, NOT 등의 논리게이트로 구성되는 논리회로로 변환될 수 있다.
- 부울함수에서 문자(iteral)
  - 정상형(X) 또는 보수형($\bar{X}$)으로 나타난 논리변수로 논리회로 구현 시 게이트 입력으로 스인다.
- 부울함수에서 항(term)
  - 문자가 논리 연산자에 의해 서로 연결된 것으로, 논리회로 구현 시에 게이트로 나타난다
- 부울함수 간소화의 목적
  - 논리회로를 구성하는 게이트 수와 게이트의 입력을 나타내는 변수의 수를 줄여 논리회로를 간소화
- 1) 항 결합
  - 2개의 항을 결합하여 하나의 항으로 만드는 방법
- 2) 문자 소거
  - 중복된 문자를 제거
- 3) 중복항 첨가
  - 주어진 함수식의 의미가 변하지 않도록 주의하면서 적절한 항을 함수식에 첨가
    - 즉, $X \cdot \bar{X}$를 부울식에 더하거나, $X+\bar{X}$를 부울식에 곱하거나, 또는 $X \cdot Y$를 X에 더하거나, X+Y를 X에 곱하는 방식
      - 첨가된 항을 이용하여 다른 항을 결합 또는 제거할 수 있음
  - 예제 3.1
  - 예제 3.2 / 3.3

4. 부울함수의 보수
- 부울함수 F의 보수는 $\bar{F}$이며, 이는 F의 값 0은 1로, 1은 0으로 바꿈으로써 얻을 수 있다. 또한 드모르간의 법칙을 이용하여 대수적으로 얻을 수도 있다. 이 법칙은 부울식의 보수를 취하려면 AND와 OR 연산을 서로 바꾸고 각 변수의 보수를 취하면 된다는 사실을 일반화한 것이다.
- 예제
  - 3.4
  - 3.5
- 함수의 보수를 쉽게 구하는 방법
  - 논리연산자인 ·(AND)는 +(OR)로, +(OR)는 ·(AND)로 바꾸어 쌍대형태를 취한 후, 각 문자의 보수를 취하는 것이다. 이 방법은 일반화한 드모르간의 법칙을 이용하는 방법이다.
  - 예제
    - 3.6
    - 3.7

3. 부울함수의 정규형 및 표준형 [84]

1. 정규형
- 개요
  - 부울함수의 정규형(canonical form)
    - 부울함수를 최소항의 합(sum of minterm)이나 최대항의 곱(product of maxterm)으로 표현한 것
  - 비정규형(non-canonical form)
    - 정규형 이외의 함수
  - 주어진 진리표 하나에 대응하는 부울함수는 무수히 많지만, 부울함수의 정규형 중 최소항의 합형태와 최대항의 곱형태의 부울함수는 진리표와 1:1 관계로 각각 오직 하나만 존재한다.
- 1) 최소항과 최대항
  - 논리변수는 정상형(X) 또는 보수형($\bar{X}$)으로 나타낸다.
    - 2개의 논리변수 X, Y가 논리곱(AND) 연산으로 묶인 경우, 각 변수는 정상형 또는 보수형으로 나타낼 수 있다.   다시 말해, XY, $X\bar{Y}$, $\bar{X}Y$, $\bar{X}\bar{Y}$와 같이 네 가지 조합이 가능하다.
      - n개의 변수는 2^n개의 최소항을 형성한다
        - 0부터 2^n - 1 까지의 2진수 조합이 n개의 변수 밑에 기록된다.
  - 세 변수에 대한 최소항과 최대항 [85]
  - 최소항(minterm)
    - 각 변수의 문자 1개씩 모두 n개의 문자를 논리곱항으로 그 결과가 논리-1인 경우
    - 각 최소항은 n개의 변수에 관한 문자(정상형이든 보수형이든 관계없이)들을 논리곱(AND)항으로 결합하되, 그 결과가 논리-1이 되도록 결합한다.

- 각 최소항에 대한 기호는 m_j 형태로 표현
  - j는 최소항에 해당하는 2진수를 10진수로 표시한 것이다
  - 최소항은 m_j로 표시하는데, 만일 논리변수가 X, Y, Z인 경우 m_5는 $X \cdot \bar{Y} \cdot Z$를 의미한다. 왜냐하면 j=5는 2진수 101을 의미하고, 문자들의 논리곱이 1이 되려면 X, $\bar{Y}$, Z의 세 문자가 논리곱으로 묶여야 하기 때문이다.

- 최대항(maxterm)
  - 각 변수의 문자 1개씩 모두 n개의 문자로 구성되는 논리합합으로 그 결과가 논리-0인 경우
  - n개의 변수는 2^n개의 최대항(maxterm)을 형성한다.
  - n개의 변수의 값이 이루는 각자의 2진수 조합에 대하여 각 변수에 관한 문자들을 논리합(OR)으로 결합하여, 그 결과가 논리 0이 되도록 결합하면 해당 최대항을 얻을 수 있다.
  - 각 최대항은 M_j 형태로 나타냄
    - 여기서 j는 최대항에 해당하는 2진수를 10진수로 표시한 것이다.
    - 최대항은 M_j로 표시하는데, 만일 논리변수가 X, Y, Z인 경우 M_5는 $\bar{X}+Y+\bar{Z}$를 의미한다. 왜냐하면 j=5는 2진수 101을 의미하고, 문자들의 논리합이 0이 되려면 $\bar{X}$, Y, $\bar{Z}$의 세 문자가 논리합으로 묶여야 하기 때문이다.
- 주어진 변수값의 2진수 조합에 대해 최소항과 최대항은 서로 보수관계에 있다.
  - 즉 $\bar{m}_j = M_j$ 인 것이다
    - 드모르간 법칙으로 쉽게 증명 가능
- 진리표를 오직 하나의 부울함수(최소항의 합형태의 정규형)으로 표현하는 방법 [85]
  - 진리표에서 출력이 1이 되는 변수들의 각 조합에 대해서 최소항을 형성하고 각각의 최소항을 논리합(OR)으로 묶으면 해당 진리표를 만족하는 정규형 부울함수를 얻을 수 있다.
    - 예를 들어, <표 3.4>의 진리표를 살펴보자. F가 1이 되려면 X, Y, Z의 2진 조합이 001, 100, 111 중 하나면 된다. 이것은 세 가지 경우가 OR 관계인 까닭이다.
    - 그런데 2진 조합 각각은 X, Y, Z가 서로 AND 관계에 있으며, 따라서 F는 $\bar{X} \cdot \bar{Y} \cdot Z$이거나 $X \cdot \bar{Y} \cdot \bar{Z}$이거나 $X \cdot Y \cdot Z$일 때 1의 값을 갖는 것이고 이것을 논리합으로 정리하면
      - $F(X,Y,Z) = \bar{X} \cdot \bar{Y} \cdot Z + X \cdot \bar{Y} \cdot \bar{Z} + X \cdot Y \cdot Z$
      - 부울함수의 정규형 중 최소항의 합 형태로 표현된 것으로 그 표현이 유일하다.
      - 즉, 진리표 하나에 대응되는 최소항의 곱형태의 부울함수는 오직 하나밖에 없다.
- 부울함수의 보수
  - 부울함수의 보수는 진리표에서 함수를 0으로 하는 최소항을 얻은 다음 이들의 OR 연산을 취하면 부울함수의 보수를 구할 수 있다.
    - <표 3.4>의 진리표를 만족하는 부울함수 F의 보수는
    - F의 보수를 다시 보수화하면 F를 얻을 수 있다.
      - 진리표 하나에 대응되는 최대항의 곱형태의 부울함수는 오직 하나다.
- 부울함수의 정규형
  - 진리표를 부울함수로 표현하는 데 두 가지 유일한 표현형태, 즉 (1) 최소항의 합형태와 (2)최대항의 곱형태가 부울함수의 정규형이다.

2) 최소항의 합

- 부울함수에서 n개의 2진 변수는 최대 2^n개의 서로 다른 최소항(AND 연산)으로 구성할 수 있으며, 구성된 최소항을 논리합(OR 연산)으로 결합하여 해당 부울함수를 표시할 수 있다.
- 때때로 부울함수를 최소항의 합으로 표현하는 것이 편리하다. 만일 그렇게 표현되지 않은 것은 먼저 곱의 합(AND 항의 합)으로 전개한다.
  - 그다음 각 항이 모든 변수를 포함하고 있는지 확인하되, 변수가 빠져 있으면 모든 변수가 포함되도록 필요한 조작을 한다.
  - 예를 들어, X 변수가 빠져 있는 항에는 $(X + \bar{X})$와 같은 표현으로 연산을 한다.
- 예제 3.8
  - 예제 3.8의 결과를 보면 5개의 최소항이 논리합으로 더해져 있다. 여기서, 최소항의 합 표현을 다음 기호를 이용하여 간단히 나타낼 수 있다.
    - $F = m_7 + m_6 + m_5 + m_4 + m_2$
    - $F(X, Y, Z) = \Sigma m(2, 4, 5, 6, 7)$
      - 좌변은 F가 부울변수 X, Y, Z의 함수라는 것을 의미한다.
      - Σ(시그마)는 항들이 논리합(OR)로 더해진 것을 나타낸다.
      - Σ 다음 소문자 m은 최소항을 나타내며, 그 다음에 나오는 숫자는 부울함수를 구성하는 각 최소항의 인덱스 번호이다.
      - 최소항의 인덱스 번호는 10진수로서 진수로 바꾸면 부울함수가 논리-1이 되는 부울수의 2진 조합을 알려준다.
        - 예를 들어, 첫 인덱스 번호인 2는 2진수로 바꾸면 (010)이므로 X=0, Y=1, Z=0일 때 F=1이 되는 것이다. 그런 이유로 좌변에 있는 F 다음의 괄호 속에 나열한 부울변수 X, Y, Z의 순서는 함부로 바꾸지 말아야 한다.

- 3) 최대항의 곱
  - 부울함수에서 n개의 논리변수는 최대 2^n개의 서로 다른 최대항(합항·OR 연산)으로 구성할 수 있으며, 그 중 필요한 최대항을 논리곱(AND)으로 결합함으로써 해당 부울함수를 표시할 수 있다. 주어진 부울함수를 최대항의 곱 형태로 나타내려면 먼저 합항의 곱 형태로 전개해야 한다.
  - 그리고 만일 각 합항에 변수 X가 빠져 있으면 $(X + \bar{X})$를 더해 준다. 이런 방법으로 합항에 n개의 변수가 모두 들어가면 원하는 최대항의 곱 형태를 얻을 수 있다.
  - 예제 3.9
    - [예제 3.9]의 결과를 보면 4개의 최대항이 논리곱으로 곱해져 있다. 여기서 최대항의 곱 표현을 다음 기호를 이용하여 간단히 나타낼 수 있다. (표 3.3)의 최대항 표기(M_j)를 이용하면 [예제 3.9]의 부울함수 F는 $F = M_0 \cdot M_2 \cdot M_4 \cdot M_5$로 표현할 수 있다. 또한 다음과 같이 나타낼 수 있다. $F(X, Y, Z) = \prod M(0, 2, 4, 5)$
      - 곱의 기호 ∏(파이)
        - 항들이 논리곱(AND)으로 곱해진 것
      - 대문자 M
        - 최대항, 괄호안의 숫자는 부울함수를 구성하는 최대항의 인덱스 번호
- 4) 정규형 간의 관계
  - 최소항의 합으로 표시된 함수의 보수는 원래 함수에서 제외된 최소항들의 합이다.
    - 어떤 함수를 표현하는 최소항은 함수를 1로 하는 데 비해, 그것의 보수를 표현하는 최소항은 처음 함수를 0으로 한다.
  - 드모르간 법칙을 적용하여 보수를 취하면 최대항의 곱형태로 표시되는 함수 F를 얻을 수 있음

3. 부울함수의 정규형 및 표준형 [90]

- 2. 표준형
  - 개요
    - 정규형
      - 대수식의 가장 기본적인 형태
      - 진료표로부터 바로 얻을 수 있지만, 최소항 또는 최대항에 항상 모든 변수가 포함되어 있으므로 부울함수의 간소화에는 적합하지 않다.
    - 표준형(standard form)
      - 표준형의 각 항은 하나 또는 그 이상의 문자로 구성된다.
      - 유형
        - 곱의 합(sum of products)
        - 합의 곱(product of sums)
  - 1) 곱의 합(SOP: sum of products)
    - 최소항의 합형태
      - 최소항의 합형태는 진리표로부터 직접 구할 수 있는 표준형의 부울식으로, 이 식은 최대수의 곱항으로 구성되며 각 항은 최대수의 문자를 갖는다.
        - 왜냐하면 정의에 의해서 각 최소항은 보수가 취해졌거나 취해지지 않은 함수의 모든 변수를 포함해야 하기 때문이다.
      - 논리함수를 간소화 하려면
        - 최소항의 합을 진리표로부터 구한 다음, 곱항의 수와 문자의 수를 줄일 수 있는지 여부를 따져 보아야 한다.
      - 예제
        - 부울함수를 곱의 합으로 표현한 형태
          - (식 3.7)은 3개의 곱항으로 구성
          - 첫 번째 항은 하나의 문자를, 두 번째 항은 3개의 문자를 그리고 세 번째는 2개의 문자를 가진다.
            - 곱의 합을 나타내는 식의 논리회로도는 하나의 OR 게이트와 여러 개의 AND 게이트 그룹으로 구성된다.
              - 입력변수 중 보수는 직접 이용할 수 있다고 가정하여 논리회로도에 NOT 게이트는 포함시키지 않는다.
              - OR 게이트를 수반한 AND 게이트 형식은 2단계(two-level) 구현이라는 회로구성을 갖게 된다.
        - 만일부울함수가 곱의 합형태가 아니라면, 분배법칙을 이용하여 표준형으로 바꿀 수 있다.
          - $F = AB + C(D + E)$ (식 3.8)
            - 곱의 형태가 아니다. 왜냐하면 항 (D + E)는 합의 형태이지 단일변수는 아니기 때문이다.
            - 이 식은 괄호를 제거하여 식(3.9)처럼 곱의 형으로 변환할 수 있다.
              - $F = AB + C(D + E) = AB + AC + CE$ (식 3.9)
              - 일반적으로 디지털 논리회로의 구현에서는 2단계 구현을 선호한다. 입력에서 출력까지의 신호전달을 위한 지연시간을 최소화할 수 있기 때문이다.
        - 3.10

2) 합의 곱(POS : product of sums)

- 대수적으로 표현한 부울함수의 또 다른 표준형은 합의 곱형태로 합항들이 논리곱형태로 구성된다.
  - 이 때 각 논리합의 항은 논리변수 전체 수보다 적은 수의 문자를 가질 수 있다.
- 식 3.10
  - 각각 하나, 둘, 그리고 세 개의 문자로 구성된 3개의 합항을 갖는다.
  - 합의 곱을 나타내는 식의 논리회로도는 하나의 AND 게이트와 합항들을 구현하는 (단일문자로 구성된 항을 제외한) OR 게이트 그룹으로 구성된다.
  - 2단계(two-level) 구현
    - 따라서 (식 3.10)을 2단계로 구현하면 [그림 3.19]와 같은 논리회로도를 얻게 된다.
  - 만약 부울함수가 합의 곱 형태가 아니라면 분배법칙[공식 (15)]을 이용하여 표준형으로 바꿀 수 있다. [예제 3.11]