closure(폐쇄) 뭔지 공부해요기

2.5장 참고하면

truth table을 보고 바로 gate를 그릴수는 없음 그렇기 때문에 수식으로 표현을 해주어야 하고 그 수식을 이제 계산을 통해 gate를 이용한 회로를 그릴 수 있음

duality:

이건 이제 2진 연산자 (and, or) 간에, 단위원(0,1)간에를 상호 바꾸어 주는것 교재 2.4절 (a)를 (b)로 바꾸어주는 것이다.

truth table을 사용하여

x+0 = x

1+0 = 1

0+0 = 0

이런 식으로 교재에 있는 2.4절 예시를 확인해 줄 수 있음

|  |
| --- |
|  |

|  |
| --- |
|  |

F(x,y,z)

variable : x, y ,z 경우의 수 8개

literals: x , x', y ,y', z, z'(literal의 개수가 해당 회로의 복잡도를 결정함 그러므로 적으면 적을 수록 좋음) 걍 같아도 상관없이 개수로 쳐줌

ex) x+w'y+wxy'z literal 개수 8개(x 공통되었지만 따로 쳐줌)

ab'+bc'd+ad'+e' = 8 literals

product term: one or more literals connected by AND(\*)

minterm: a product term that includes all the variables, either complement or not

ex) for 4 variables: x, w, y, z (minterm은 그 값이 1이어야 함)

w'xyz , wxyz: minterm

wy'z: non minterm

Sop = sum of products

- one or more product connected by or

ex) w'x'yz' + wx'y'z' +wxyz + wx'yz (4 product terms)

x + w'y +wxy'z (3 product terms)

x'+ y + z (3 product terms)

canonical sum or canonical SOP = sum of minterm

그니깐 or로 이루어지고 모든 variable이나 역수가 있어야함 (all \*)+

canonical SOP:얘는 모든 변수 다포함하고 or로 구성되어야 함

canonical sum: 얘는 모든 변수는 다 포함 안해도 됨 근데 이제 적어도 하나 이상의 변수나 그 변수의 보수가 or로 연결되어야 함

- w'x'yz' + wx'y'z' +wxyz + wx'yz -canonical SOP

minimum sum of products

1) Sop 중에서 minimum을 찾는 방법은 product term을 세서 그놈이 최소인것이 minimum sop이다

2) 만약 product term이 동일하다면 literal의 개수를 찾는다

ex) w'x'yz' + wx'y'z' +wxyz + wx'yz (4 product terms)

x + w'y +wxy'z (3 product terms)

x'+ y + z (3 product terms)

여기서 2,3이 product term이 3개 이므로 minimum이다. 근데 2는 literal이 4개 3은 literal이 3개이므로 3번이 minimum sum of products이다.

sumterm = one or more literals connected by OR(+)

ex) a+b'+c, b'

standard sum term or maxterm = a sum term that includes all the variables, either complemented or not

ex)

for a 4 variables, w, x, y and z

w' +x+y+z' , w+x+y+z: maxterms

x+y'+z: not maxterm but sum term

product of sum(POS)

one or more sum terms connected by AND(\*)

ex)

-(w+x)(w+y) => 2sum terms

-w(z+y) => 2 sum terms

- w 1sum terms

Canonical and Standard Forms

|  |
| --- |
|  |

ex)

|  |
| --- |
|  |
| f2= x'yz+xy'z+xyz'+xyz = m3+m5+m6+m7 (x,y,w,z 은 1, x',y',w',z' 는 0을 의미)  f1 = (x+y+z)(x+y'+z)(x+y’+z’)(x'+y+z')  (x'+y'+z) = M0M2M3M5M6  f2= (x+y+z)(x+y+z‘)(x+y'+z)(x'+y+z)  = M0M1M2M4 |

Sum of Minterms

ex) F라는 함수가 A,B,C라는 변수를 가질 때 F = A+B'C 의 sum of minterms 를 구하여라

A = A(B+B') = AB+AB' = AB(C+C') + AB'(C+C') = ABC+ABC'+AB'C+AB'C'

|  |  |
| --- | --- |
|  | A = A(B+B') = AB+AB'  = AB(C+C') + AB'(C+C')  = ABC+ABC'+AB'C+AB'C'  B'C = B'C(A+A') = AB'C + A'B'C  F = A+b'C  = ABC+ABC'+AB'C+AB'C' + AB'C + A'B'C  = m1 + m4 + m5+m6+m7  = ∑(1, 4, 5, 6, 7) |

Product of maxterms

Express the Boolean function F=xy+x'z in a product of maxterm form.

|  |  |
| --- | --- |
|  | F = xy +x'z = (xy+x')(xy+z)[absorption?]  (그냥 앞에 걸로 뒤에거 두개 분해하고 묶어주면 됨)  = (x+x')(y+x')(x+z)(y+z) = (y+x')(x+z)(y+z)  x' + y= x' + y + zz'= (x'+y+z)(x'+y+z')  x + z= x + z + yy'= (x+y+z)(x+y'+z)  y + z= y + z + xx'= (x+y+z)(x'+y+z)  F = (x+y+z)(x+y'+z)(x'+y+z)(x'+y+z')  = M0M2M4M5  F(x, y, z) = ∏(0, 2, 4, 5) |

|  |
| --- |
|  |
|  |
|  |
|  |

associative 하다: 결과가 똑같다

exclusive or로 만약 회로가 구성되었다면 회로의 결과 값은 1의 개수가 홀수면 결과값은 1이고 1의 개수가 짝수면 결과값은 0이다.

2.9 INTEGRATED CIRCUITS[2.9장 전부 외울것]

*- Levels of Integration*

SSI(Small-scale integration) : few gates (10개 이하)

MSI(Medium-scale integration) : 10 to 1000 gates

LSI(Large-scale integration) : thousands of gates

ex) processors, memory chips, programmable logic devices

VLSI(Very large-scale integration) : millions of gates

ex) large memory arrays, complex microcomputer chips

*-Digital Logic Families*

TTL(Transistor-transistor logic) : standard form

ECL(Emitter-coupled logic) : high speed operation

MOS(Metal-oxide semiconductor) : high component density

CMOS(Complementary metal-oxide semiconductor) :

low power consumption, recently used for VLSI design

*-Parameters of logic families*

**\*Fan-out: the number of standard loads that the output of a typical gate can drive without impairing its normal operation**

**(출력이 다음 gate의 입력으로 몇개까지 들어갈 수 있느냐)**

**\*Fan-in: the number of inputs available in a gate**

**(입력이 gate에 몇개까지 들어갈 수 있느냐)**

**\*Power dissipation : the power consumed by the gate**

**(하나의 gate마다 소모되는 전력)**

**\*Propagation delay : the average transition delay time for a signal to propagate from input to output**

**(gate안에 nput이 들어와서 output이 나올 때까지 걸리는 시간 즉 지연된다)**

**(그러므로 gate가 적을 수록 걸리는 시간이 적어짐)**

*- Computer-Aided Design of VLSI Circuits*

CAD(Computer-aided design)

ASIC(Application-specific IC), FPGA(Field-programmable gate

array), PLD(programmable logic device), Full-Custom IC

HDL(Hardware Description Language) : VHDL and Verilog

FPGA: 칩이 있으면 이것의 기능이 정해져 있음

근데 FPGA는 현장에서 코딩이 가능하여 칩의 기능을 HDL에 따라서 작동하게 할 수 있음