Midterm1 Review

Jan 24

Armita

Outline

- Boolean Algebra
  - Axioms
    - closure, Identity elements, complements, commutativity, distributivity
  - theorems
    - Associativity, Duality, De Morgan,
- Consensus theorem
- Shannon Expansion
- Canonical SOP & POS, minterms, maxterms
- Implicant, Prime Implicant, Essential Prime Implicant
- Implicates, Prime Implicates, Essential Prime Implicates
- Minimized two level SOP/POS using Kmaps

Boolean Algebra

Boolean Algebra is a multiple valued logic {a0, a1, a2, ...an}, defined over two operations {op1, op2}, that satisfies the following properties:

- It is closed with respect to each of the operations.
- Each operation has an identity element:
  - there exist x such that: A op1 x = A , for each A ∈ {a0, a1, a2, ...an}
    - we call x the identity element of op1
  - there exist y such that: A op2 y = A , for each A ∈ {a0, a1, a2, ...an}
    - we call y the identity element of op2

for example: 1 AND X = X → "1" is the identity element of AND operation.

$x \in \{0,1\}$

(in switching algebra) 0 OR X = X → "0" is the identity element of OR operation.

complements

- Each Element (M) has a unique complement element (N), defined as follows:
  - M op1 N = Identity element of op2
  - M op2 N = Identity element of op1
    - for each M from {a0, a1, a2, ...an} we can find a unique N from {a0, a1, a2, ...an} that satisfies the above equalities and we call them complements of each other.
    - We cannot have Boolean algebra that has odd number of values!

for example: (in switching algebra)

0 OR 1 = 1 = Identity element of AND operation

0 AND 1 = 0 = Identity element of OR operation

0 and 1 are complement of each other!

Axioms

- **Commutativity:**

for each X,Y ∈ {a0, a1, a2, ...an} :

duality

X op1 Y = Y op1 X

X op2 Y = X op2 Y

switching Algebra

for each X,Y ∈ {0,1} :

X AND Y = Y AND X

X OR Y = X OR Y

- **Distributivity:**

for each X,Y, Z ∈ {a0, a1, a2, ...an} :

duality

X op1 (Y op2 Z) = (X op1 Y) op2 (X op1 Z)

X op2 (Y op1 Z) = (X op2 Y) op1 (X op2 Z)

switching Algebra

for each X,Y, Z ∈ {0,1} :

X AND (Y OR Z) = (X AND Y) OR (X AND Z)

X OR (Y AND Z) = (X OR Y) AND (X OR Z)

Knowing these Axioms, we can prove useful theorems for Boolean Algebra, ex:
De morgan law, Consensus theorems ...

Consensus Theorem

consensus theorem : AB + A'C + BC = AB + A'C [BC is called the consensus term of AB and A'C]

In a general form: AN + A' M + MN = AN + A'M where M and N are boolean expressions, for example:

abcd + a'ce + bcde = abcd + a'ce [(bcd)(ce) is the consensus term, so its redundant.]

abc + a' (c + d) + bc + bcd = abc + a'(c+d)

[(bc)(c+d) is the consensus term (which is equal to bc +bcd), so its redundant.]

Consensus Theorem

In the dual form :

consensus theorem: (A+B) (A' + C) (B + C) = (A+B) (A'+C)   [(B+C) is called the consensus term of (A+B) and (A'+C)]

In a general form: (A + N)(A' + M)(M+N) = (A+N)(A'+M) where M and N are boolean expressions, for example:

(a + b + c + d) (a' + c + e) (b + c + d + e) = (a + b + c + d) (a' + c + e)

[(b+c+d)+(c+e) is the consensus term of (a + b + c + d) and (a' + c + e) so its redundant.]

Shannon Expansion

$f(x, y, z) = x.f(1, y, z) + x'.f(0, y, z)$

in " f " replace x with 1 and x' with 0   in " f " replace x with 0 and x' with 1

$= y.f(x, 1, z) + y'.f(x, 0, z)$

$= z.f(x, y, 1) + z'.f(x, y, 0)$

only for switching algebra!!

example: simplify the given function:

$f(x, y, z) = xyz + x'y + z'y$

On which variable should be apply the Shannon expansion theorem?

Shannon Expansion

example: simplify the given function:

$f(x, y, z) = xyz + x'y + z'y$

Lets choose x !

$= x.\ f(1, y, z) + x'.f(0,y,z)$

$= x\ .\ (yz + 0 + z'y) + x'\ .\ (\ 0 + y + z'y)$

$= x\ .\ (y\ (z+z'))\ \ + x'\ .\ (y\ (1+z'))$

$= x.\ y\ \ + x'.\ y$

$= y\ (x+x')$

$= y$

Shannon Expansion

example: simplify the given function:

$$f(x, y, z) = xyz + x'y + z'y$$

what if we choose z ?!

$$= z.\ f(x, y, 1) + z'.f(x,y,0)$$

$$= z\ .\ (xy + x'y + 0) + z'\ .\ (\ 0 + x'y + y)$$

$$= z\ .\ (y\ (x+x'))\ \ + z'\ .\ (y\ (x'+1))$$

$$= z.y + z'.\ y$$

$$= y\ (z+z')$$

$$= y$$

Shannon Expansion

$f(x, y, z) = (x + f(0, y, z)) \, . \, (x' + f(1, y, z))$

in " f " replace x with 0 and x' with 1 in " f " replace x with 1 and x' with 0

$= (y + f(x, 0, z)) \, . \, (y' + f(x, 1, z))$

$= (z + f(x, y, 0)) \, . \, (z' + f(x, y, 1))$

example: simplify the given function:

$f(x, y, z) = (x + y + z')\,(x' + y' + z)\,(z + y)$

Shannon Expansion

example: simplify the given function:

$$f(x, y, z) = (x + y + z')\,(x' + y' + z)\,(z + y)$$

$$= (x + f(0, y, z))\ .\ (x' + f(1, y, z))$$

$$= (x + (y+z')\,(1)\,(z+y)\,)\ .\ (x' + (1)\,(y'+z)\,(z+y)\,)$$

$$= (x + y + z'z\,)\ .\ (x' + z + y'y)$$

$$= (x + y + 0\,)\ .\ (x' + z + 0\,)$$

$$= (x+y)\,(x'+z)$$

Definitions:

literal → a variable in the direct or complemented form: x , x' , y , y'

product term → AND of some literals: (xy'z) , (xy) can differ in their size

sum term → OR of some literals: (x + y' + z) , (y+z) , can differ in their size

Minterm

minterm → a product term that has all the variables in either the direct form or the complement form. For a n-variable system we have *2^n* minterms and all have the same size (one AND gate with n input): {a, b,c,d} → abcd, abcd', abc'd, …, a'b'c'd'

Each minterm is equal to value " one " only for a unique input combination and is " zero " otherwise:

" abcd " will be one if and only if (a,b,c,d) = (1,1,1,1)

" abcd' " will be one if and only if (a,b,c,d) = (1,1,1,0)

" abc'd " will be one if and only if (a,b,c,d) = (1,1,0,1)

…..

" a'b'c'd' " will be one if and only if (a,b,c,d) = (0,0,0,0)

Maxterm

maxterm → a sum term that has all the variables in either the direct form or the complement form. For a n-variable system we have *2^n* maxterms and all have the same size (one OR gate with n input): {a,b,c, d} → (a+b+c+d) , (a+b+c+d') , (a+b+c'+d) , …, (a'+b'+c'+d')

Each maxterm is equal to value " zero " only for a unique input combination and is " one " otherwise:

" a+b+c+d " will be zero if and only if (a,b,c,d) = (0,0,0,0)

" a+b+c+d' " will be zero if and only if (a,b,c,d) = (0,0,0,1)

" a+b+c'+d " will be zero if and only if (a,b,c,d) = (0,0,1,0)

….

" a'+b'+c'+d' " will be zero if and only if (a,b,c,d) = (1,1,1,1)

Canonical form

Based on the definition of minterms and maxterms, we can represent any switching function with two canonical form:

canonical sum of products: Sum of its minterms (i.e ORing all places where the function is one)

canonical product of sums: Product of its maxterms (i.e ANDing all places where the function is zero)

canonical form

Example: Show the canonical forms for a switching function that receives 3 binary inputs {a,b,c} and the output {F} is one if and only if the number of ones in the input are even:

- draw the truth table representing F(a,b,c)
- find the minterms → write canonical SOP form
- find the maxterms → write the canonical POS form

3 inputs {a,b,c} → the truth table has 2^3 = 8 rows:

a	b	c	F	
0	0	0	1	a'b'c'
0	0	1	0	a+b+c'
0	1	0	0	a+b'+c
0	1	1	1	a'bc
1	0	0	0	a'+b+c
1	0	1	1	ab'c
1	1	0	1	abc'
1	1	1	0	a'+b'+c'

canonical SOP

F (a,b,c) = a'b'c' + a'bc + ab'c + abc'

$= \sum m\,(0, 3, 5, 6)$

= m0 + m3 + m5 + m6

canonical POS

F (a,b,c) = (a+b+c') (a+b'+c) (a'+b+c) (a'+b'+c')

$= \prod M\,(1, 2, 4, 7)$

= M1 . M2 . M4 . M7

Logic Minimization

Representing a function in the canonical form is not efficient. Hence, we are looking for ways to merge two or more minterms (maxterms) together in order to get a smaller product (sum) term which is more efficient.

Adjacency: two minterms (maxterms) are adjacent, if they differ only in one variable: abcd and ab’cd are adjacent. (a+b+c) and (a+b+c’) are adjacent.

We can merge each Adjacent pair into one term !

abcd + ab’cd = acd (b+b’) = acd

(a+b+c)(a+b+c’) = (a+b) + (cc’) = a+b

K-maps!

Searching for adjacent pairs are the basis of logic minimization.

Karnaugh maps present a visual way to efficiently look for this pairs!

All neighbour cells in a Kmaps are adjacent.   Dont forget the edges!!

d   c

ab\cd	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

b

a

We can use Kmaps for 5 or 6 variables function.
In a 5 variable kmap, each cell has 5 neighbours and in a 6 variable kmap each cell has 6 neighbours.

Definitions:

on-set : set of all the “ones” in the function

off-set : set of all the “zeros” in the function

Implicant: A product term that has non-empty intersection with on-set and does not intersect with the off-set. It can also contain “dont cares”

- Being a product term means that it should include 1 minterm, or 2 minterms, or 4, …(powers of two)... . It CANNOT have 3 minterms!

Definitions:

Prime Implicant: Is an Implicant which is not a proper subset of any other Implicant. i.e you cannot find any other implicant that completely covers a prime implicant. For logic minimization we use Prime Implicants.

Essential Prime Implicant: A Prime Implicant that has at least one element from the on-set that is cannot be covered by any other Prime Implicant. By this definition, these Implicants are essential and will show up in all minimal SOPs.

Minimal SOP

- Draw the kmap representing the given function
- Find all the Prime Impilcants using the on-set.
- Find all the Essential Prime Implicants.
- Write all the Essential Prime Implicants in the final cover,
- For the ones in the on-set which are not covered yet, search through the remaining non-essential prime implicants that will cover the remaining ones in the map. By trial and error at this step you can find all the minimal covers.

Example

d c

ab\cd	00	01	11	10
00	0 1	1	3	2 1
01	4 1	5 1	7 1	6 1
11	12	13 1	15 X	14
10	8	9 1	11	10

b

a

The empty cells are “0”.

Prime Implicants:

?

Essential Prime Implicants:

?

Minimal SOP:

?

Example

ab\cd	00	01	11	10
00	0 1	1	3	2 1
01	4 1	5 1	7 1	6 1
11	12	13 1	15 X	14
10	8	9 1	11	10

The empty cells are "0".

Is ∑m (0,4) a Prime Implicant?

NO!! it can be covered by ∑m (0,4,2,6)

How about ∑m (5,9,13)?

NO!! Its not a product term, we cannot have product terms with 3 minterms!

Example

d c

b a

ab\cd	00	01	11	10
00	0 1	1	3	2 1
01	4 1	5 1	7 1	6 1
11	12	13 1	15 X	14
10	8	9 1	11	10

The empty cells are "0".

Prime Implicants:

$\sum m(0,2,4,6) = a'd'$

$\sum m(4,5,6,7) = a'b$

$\sum m(5,7,13,15) = bd$

$\sum m(9,13) = ac'd$

Essential Prime Implicants:

$\sum m(0,2,4,6) = a'd'$

$\sum m(9,13) = ac'd$

Minimal SOP:

?

Example

ab\cd	00	01	11	10
00	0 1	1	3	2 1
01	4 1	5 1	7 1	6 1
11	12	13 1	15 X	14
10	8	9 1	11	10

The empty cells are "0".

Prime Implicants:
∑ m (0,2,4,6) = a'd'
∑ m (4,5,6,7) = a'b
∑ m (5,7,13,15) = bd
∑ m (9,13) = ac'd

Essential Prime Implicants:
∑ m (0,2,4,6) = a'd'
∑ m (9,13) = ac'd

Minimal SOP: (#1)

F (a,b,c,d) = a'd' + ac'd + a'b

Example

d c

ab\cd	00	01	11	10
00	0 1	1	3	2 1
01	4 1	5 1	7 1	6 1
11	12	13 1	15 X	14
10	8	9 1	11	10

b a

The empty cells are “0”.

Prime Implicants:

∑ m (0,2,4,6) = a’d’

∑ m (4,5,6,7) = a’b

∑ m (5,7,13,15) = bd

∑ m (9,13) = ac’d

Essential Prime Implicants:

∑ m (0,2,4,6) = a’d’

∑ m (9,13) = ac’d

Minimal SOP: (#2)

F (a,b,c,d) = a’d’ + ac’d + bd

Example

ab\cd	00	01	11	10
00	0 1	1	3	2 1
01	4 1	5 1	7 1	6 1
11	12	13 1	15 X	14
10	8	9 1	11	10

The empty cells are "0".

Prime Implicants:

$\sum m\,(0,2,4,6) = a'd'$

$\sum m\,(4,5,6,7) = a'b$

$\sum m\,(5,7,13,15) = bd$

$\sum m\,(9,13) = ac'd$

Essential Prime Implicants:

$\sum m\,(0,2,4,6) = a'd'$

$\sum m\,(9,13) = ac'd$

Minimal SOP:

$F(a,b,c,d) = a'd' + ac'd + a'b$

OR

$F(a,b,c,d) = a'd' + ac'd + bd$

Definitions:

Another approach is covering the “zeros” and finiding the minimal POS form:

Implicate: A sum term that has non-empty intersection with off-set and does not intersect with the on-set. It can also contain “dont cares”

- Being a sum term means that it should include 1 maxterm, or 2 maxterms, or 4, or ...(powers of two)... It CANNOT have 3 maxterms!

Definitions:

Prime Implicate: Is an Implicate which is not subset of any other Implicate. i.e you cannot find any other Implicate that completely cover a Prime Implicate. For logic POS minimization we use Prime Implicates.

Essential Prime Implicate: A Prime Implicate that has at least one element from the off-set that is cannot be covered by any other Prime Implicate. By this definition, these Implicates are essential and will show up in all minimal POSs.

Minimal POS

- Draw the kmap representing the given function
- Find all the Prime Impilcate using the off-set.
- Find all the Essential Prime Implicates.
- Write all the Essential Prime Implicates in the final cover,
- For the zeros in the off-set which are not covered yet, search through the remaining non-essential Prime Implicates that will cover the remaining zero in the map. By trial and error at this step you can find all the minimal covers.

Example

ab\cd	00	01	11	10
00	0 0	1	3 0	2 0
01	4 0	5	7 0	6 0
11	12	13 0	15 X	14
10	8	9	11 0	10

The empty cells are “1”.

Prime Implicates:

∏M(0,2,4,6) = a+d

∏M (3,5,7,11) = c’+d’

∏M (2,3,4,6) = a+c’

∏M(13,15) = a’+b’+d’

Essential Prime Implicates:

∏M(0,2,4,6) = a+d

∏M (3,5,7,11) = c’+d’

∏M(13,15) = a’+b’+d’

Minimal POS:

F (a,b,c,d) = (a+d)(c’+d’)(a’+b’+d’)

Boolean Algebra VS. K-maps

F (a,b,c) = c'b + a'b + a'b'c + a'bc

c\ab	00	01	11	10
0	0	1	1	0
1	1	1	0	0

Boolean Algebra VS. K-maps

F (a,b,c) = c'b + a'b + a'b'c + a'bc

adjacent

= c'b + a'b + a'b'c + a'bc

= c'b + a'b + a'c (b'+b)

= c'b + a'b + a'c

not prime!

c\ab	00	01	11	10
0	0	1	1	0
1	1	1	0	0

not prime!

c\ab	00	01	11	10
0	0	1	1	0
1	1	1	0	0

Boolean Algebra VS. K-maps

$F(a,b,c) = c'b + a'b + a'b'c + a'bc$

$= c'b + a'b + a'b'c + a'bc$

$= c'b + a'b + a'c\,(b'+b)$

$= \underline{c'b} + a'b + a'c$ (consensus)

$= c'b + a'c$

essential essential

c\ab	00	01	11	10
0	0	1	1	0
1	1	1	0	0

c\ab	00	01	11	10
0	0	1	1	0
1	1	1	0	0

QUESTIONS?

GOOD LUCK!