Lecture 3

The canonical form of combinatorial circuits and its minimisation

Combinatorial Circuits

are those in which variables on the “Right Hand Side” RHS do not appear on the “Left Hand Side” LHS:

$R = (A+B)\bullet C$

$P = (Q\bullet S')+(T+C)'$

but not

$R = (A+R)\bullet(T+C)'$

This means there is no cycle in a combinatorial circuit!

Canonical Forms

A canonical form is a standard way of writing an equation—a shape or form of equation.

Canonical forms are useful for example in algorithms for logic gates such as logic minimisation and checking if boolean expressions are equal.

Minterms

One common canonical form is a sum of minterms.

A minterm is simply a product term (ie a conjunction) containing each input (RHS) variable or its negation:

eg (for 3 variables A, B, C):

$A \bullet B \bullet C$, $A \bullet B' \bullet C'$, $A' \bullet B \bullet C$, $A \bullet B \bullet C'$

but not $B' \bullet C'$

The Sum of Products

The canonical form is simply a sum of minterms (ie a disjunction of conjuncts):

$R = A \bullet B \bullet C + A \bullet B' \bullet C' + A' \bullet B \bullet C$

Note that, for example

$R = B \bullet C + B' + A' \bullet B \bullet C$

is a valid Boolean equation, but is not in canonical form.

The canonical form and the truth table

One advantage of the canonical form is that it directly represents the truth table.

Consider the minterm: $A' \bullet B \bullet C$

This has value 1 iff $A=0$, $B=1$ and $C=1$.

So a minterm in the canonical form corresponds to a row in the truth table which has an output of 1.

Example: The majority voter

(Returns 1 iff the majority of the inputs is 1)

A	B	C	X	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	Minterm A'•B•C
1	0	0	0	
1	0	1	1	Minterm A•B'•C
1	1	0	1	Minterm A•B•C'
1	1	1	1	Minterm A•B•C

So X = A'•B•C + A•B'•C + A•B•C' + A•B•C

The canonical form derived from equations

The canonical form can also be found from equations by a process known as augmentation.

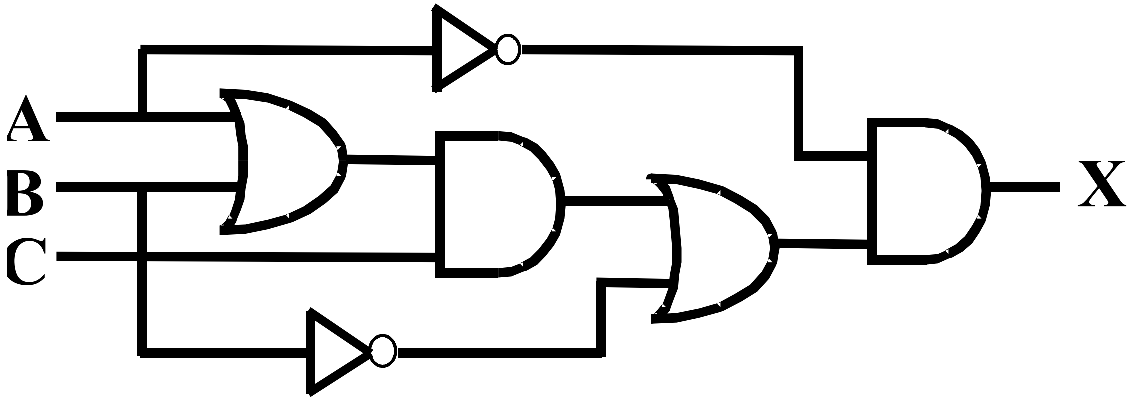
Consider $X = B \bullet C + A' \bullet B \bullet C$

We know that $(A+A') = 1$ and so we can write

$$X = (A+A') \bullet B \bullet C + A' \bullet B \bullet C$$

$$X = A \bullet B \bullet C + A' \bullet B \bullet C + A' \bullet B \bullet C$$

The canonical form derived from a circuit



By inspection the above circuit has equation:

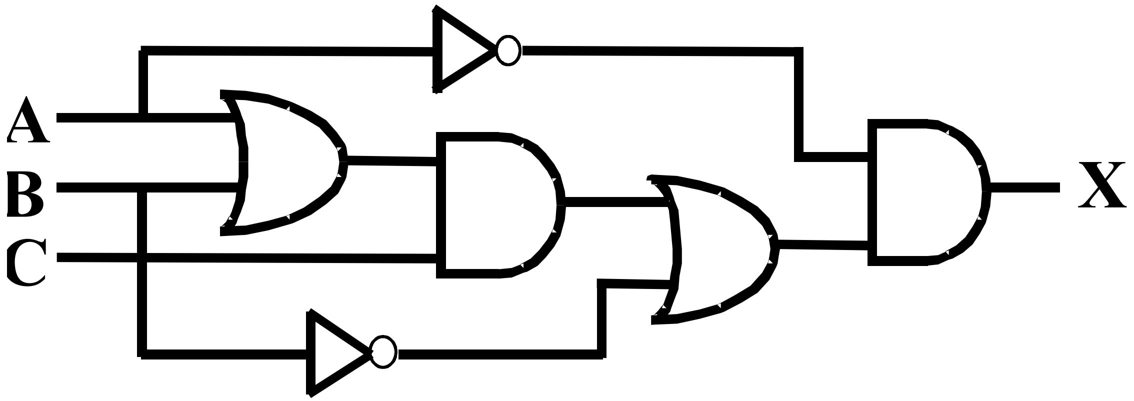
$X = A' \bullet (B' + C \bullet (A + B))$

which can be simplified by multiplying out

$X = A' \bullet B' + A' \bullet C \bullet A + A' \bullet C \bullet B$

$X = A' \bullet B' + A' \bullet C \bullet B$

Example continued



We now apply the process of augmentation to get

$X = A'\bullet B' + A'\bullet C\bullet B$

$X = A'\bullet B' \bullet(C+C') + A'\bullet C\bullet B$

$X = A'\bullet B' \bullet C + A'\bullet B' \bullet C' + A'\bullet B\bullet C$

which is the canonical form

Why use the canonical form?

Clearly the canonical form is not the smallest circuit representation.

However, it can be obtained automatically from a truth table, and easily from boolean equations and circuits.

The canonical form is used in automated design and analysis of circuits.

Minimisation

Clearly, the cheapest (and typically most reliable) circuit implementing a Boolean equation is the smallest or fastest circuit.

Thus, having obtained an equation in canonical form we seek to simplify it.

However, this is no easy task.

Consider again the majority voter

$$X = A'\bullet B\bullet C + A\bullet B'\bullet C + A\bullet B\bullet C' + A\bullet B\bullet C$$

one possible factorisation is

$$X = A'\bullet B\bullet C + A\bullet B'\bullet C + A\bullet B\bullet(C' + C)$$

$$X = A'\bullet B\bullet C + A\bullet B'\bullet C + A\bullet B$$

and that is as far as we can go

or is it ??

Further Simplification

Suppose we were to do the following

$$X = A'\bullet B\bullet C + A\bullet B'\bullet C + A\bullet B\bullet C' + A\bullet B\bullet C$$

$$X = A'\bullet B\bullet C+A\bullet B'\bullet C+A\bullet B\bullet C'+A\bullet B\bullet C+A\bullet B\bullet C+A\bullet B\bullet C$$

now we can find more factorisations

$$X = (A'+A)\bullet B\bullet C + (B'+B)\bullet A\bullet C + (C'+C)\bullet A\bullet B$$

giving

$$X = B\bullet C + A\bullet C + A\bullet B$$

which is the simplest form(!)

Karnaugh Map

Since we are not all brilliant mathematicians it is unlikely that we will find simplifications of the kind used in the last slide.

However, there is a simple graphical aid to factorisation known as the Karnaugh map.

The Karnaugh map is a truth table written out in a particular form.

Three input Karnaugh map

A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	1

This is the Karnaugh map for the majority voter

Notice that it is laid out so that only one of the variables changes in adjacent columns

Karnaugh Maps for majority circuits

A \ B	0	1
0	0	1
1	1	1

Two Input OR gate

A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Three input majority circuit

AB \ CD	00	01	11	10
00	0	0	1	0
01	0	1	1	1
11	1	1	1	1
10	0	1	1	1

Four Input Majority Circuit

Factorisations and the Karnaugh map

A \ BC	00	01	11	10
0	0	0	1	0
1	0	1	1	1

Consider the two 1s circled above.

They represent the condition in which both B and C are 1 regardless of A.

This can be represented by the equation $B \bullet C = 1$

Problem Break

BC

A	00	01	11	10
0	0	0	1	1
1	0	1	1	0

What conditions are represented by the circles above?

Problem Break

	BC			
A	٠٠	٠١	١١	١٠
٠	٠	٠	١	١
١	٠	١	١	٠

For the upper circle A is always 0 and B is always 1, so the condition is A’ • B = 1.

For the lower circle A is always 1 and C is always 1, so the term is A • C = 1

Each circle on a Karnaugh map represents a term

Term B

A \ B	0	1
0	0	1
1	1	1

Term A

Areas of the Karnaugh and their terms

AB \ CD	00	01	11	10
00	0	1	1	1
01	0	1	0	1
11	1	1	1	1
10	1	1	1	1

Term A'•B'•C

Term A'•B•C•D'

Term A

Term C'•D

Drawing circles on Karnaugh Maps

A factorisation is represented by a set of circles covering all the 1s and none of the 0s

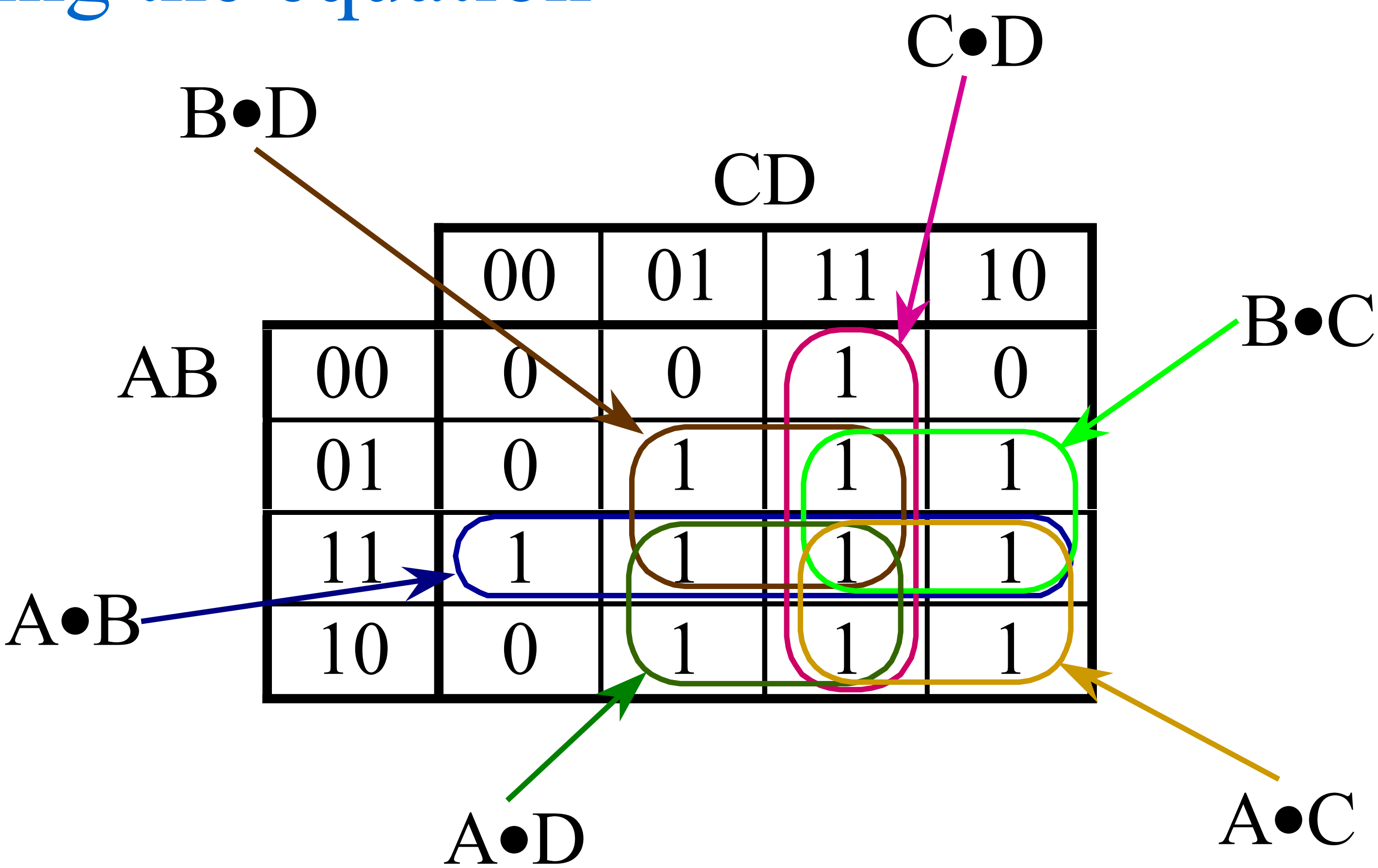
Circles on Karnaugh maps can have dimensions of 1, 2 or 4 in either direction, but never 3!

The larger the circle, the smaller the term it represents.

Covering the 1s with circles

AB \ CD	00	01	11	10
00	0	0	1	0
01	0	1	1	1
11	1	1	1	1
10	0	1	1	1

Getting the equation



$$X = A \bullet B + A \bullet D + A \bullet C + B \bullet C + C \bullet D + B \bullet D$$

Reordering the Karnaugh Map

We always ordered the Karnaugh map

00 01 11 10

We must keep this order, but we can cycle it, for example to

01 11 10 00

	00	01	11	10
00		1		
01	1			1
11			1	
10		1		1

	01	11	10	00
01			1	1
11		1		
10	1		1	
00	1			

Drawing circles round the edges

We need not redraw the map if we note that the top and bottom rows are adjacent as are the left and right columns:

	00	01	11	10
00		1		
01	1			1
11			1	
10		1		1

	01	11	10	00
01			1	1
11		1		
10	1		1	
00	1			

Don't Cares

Suppose that we are designing a circuit, and we know that certain inputs will never occur.

We call these values "don't cares" because we don't care what the circuit output is for that input.

On a Karnaugh map we write a don't care as X

Don't Cares 2

Suppose we are designing a majority voter but we know that inputs 0000 and 0100 will never occur.

The Karnaugh map becomes:

AB \ CD	00	01	11	10
00	X	0	1	0
01	X	1	1	1
11	1	1	1	1
10	0	1	1	1

Don’t Cares 3

We can choose to make the don't cares either 1 or 0 in order to make our circles as big as possible.

C•D

B

AB \ CD	00	01	11	10
00	X	0	1	0
01	X	1	1	1
11	1	1	1	1
10	0	1	1	1

$$X = A•D + A•C + B + C•D$$

A•D

A•C