COMP1071 - Digital Electronics

Karnaugh Maps

Ioannis Ivrissimtzis

email: ioannis.ivrissimtzis@durham.ac.uk

Room: MCS 2023

Slide acknowledgements: Eleni Akrida and Farshad Arvin

Overview of today's lecture

- Karnaugh Maps: a more systematic way to simplify Boolean formulae

- Example: 7-segment display driver

Input A	Input B	Input C	Input D	Output F(A,B,C,D)
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	1	0	0	1
0	0	1	1	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	0	1	1	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

$$F = \overline{A}\cdot\overline{B}\cdot\overline{C}\cdot\overline{D} + \overline{A}\cdot\overline{B}\cdot C\cdot\overline{D} + \overline{A}\cdot B\cdot\overline{C}\cdot\overline{D} + \overline{A}\cdot B\cdot\overline{C}\cdot D + \overline{A}\cdot B\cdot C\cdot\overline{D}$$
$$+\overline{A}\cdot B\cdot C\cdot D + A\cdot\overline{B}\cdot\overline{C}\cdot\overline{D} + A\cdot\overline{B}\cdot\overline{C}\cdot D + A\cdot\overline{B}\cdot C\cdot\overline{D} + A\cdot B\cdot\overline{C}\cdot D$$

Durham University

Simplifying Boolean expressions

Key to simplifying is spotting terms of the form $\boldsymbol{PA + P\,\bar{A}}$ (since this = P).

Karnaugh Maps are a graphical way of representing equations to make spotting these terms easier.

A	B	C	Y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

(a)

Y (C \ AB)	00	01	11	10
0	1	0	0	0
1	1	0	0	0

(b)

Y (C \ AB)	00	01	11	10
0	$\bar{A}\bar{B}\bar{C}$	$\bar{A}B\bar{C}$	$AB\bar{C}$	$A\bar{B}\bar{C}$
1	$\bar{A}\bar{B}C$	$\bar{A}BC$	ABC	$A\bar{B}C$

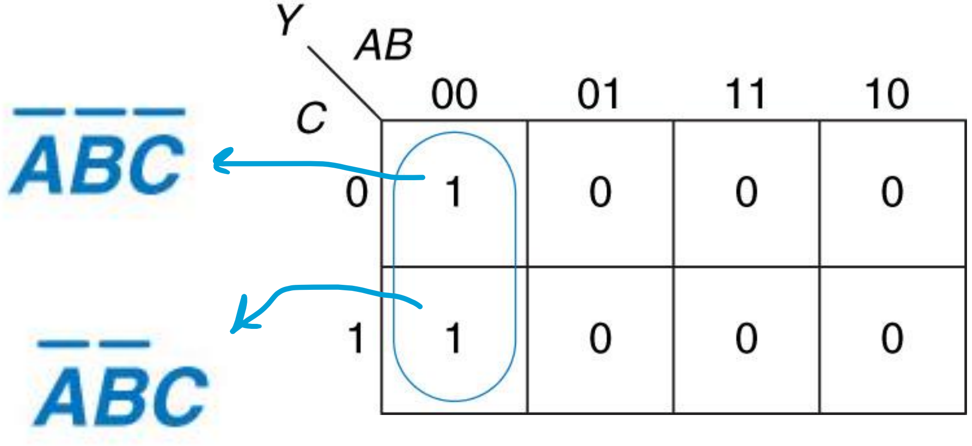
(c)

Each cell represents a minterm, and has a zero or one depending on the value of Y corresponding to that minterm. SoP form is given by adding the minterms corresponding to 1s.

The order of minterms is such that each cell differs in **the negation of exactly one variable** from its neighbours (including wrap around).

Simplifying Boolean expressions

Terms of the form $\boldsymbol{PA + P\overline{A}}$ are **neighbouring 1s** in the K-map.



Rather than writing out full SoP by taking every 1 as a term, we circle neighbouring 1s, and use the single reduced implicant for both 1s in the circle.

Full SoP: $Y = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C$

Reduced: $Y = \overline{A}\,\overline{B}$

Karnaugh Maps

1. Create the map so that neighbouring terms differ in the negation of one variable (including wrap around).
2. Circle *exactly* all **ones** in the map using as **few circles** as possible, and making each circle as **large** as possible.
3. Each circle must span a rectangular block that is a power of 2 in each dimension (i.e. 1,2,4).
4. Read off the implicants that were circled.

If a Boolean expression is minimal then it is the sum of **prime implicants**: implicants that cannot be combined with each other.

Each circle represents an implicant. Largest possible circles represent prime implicants.

Example

Truth Table

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

K-Map

Y: C \ AB	00	01	11	10
0	0	1	1	0
1	0	1	0	0

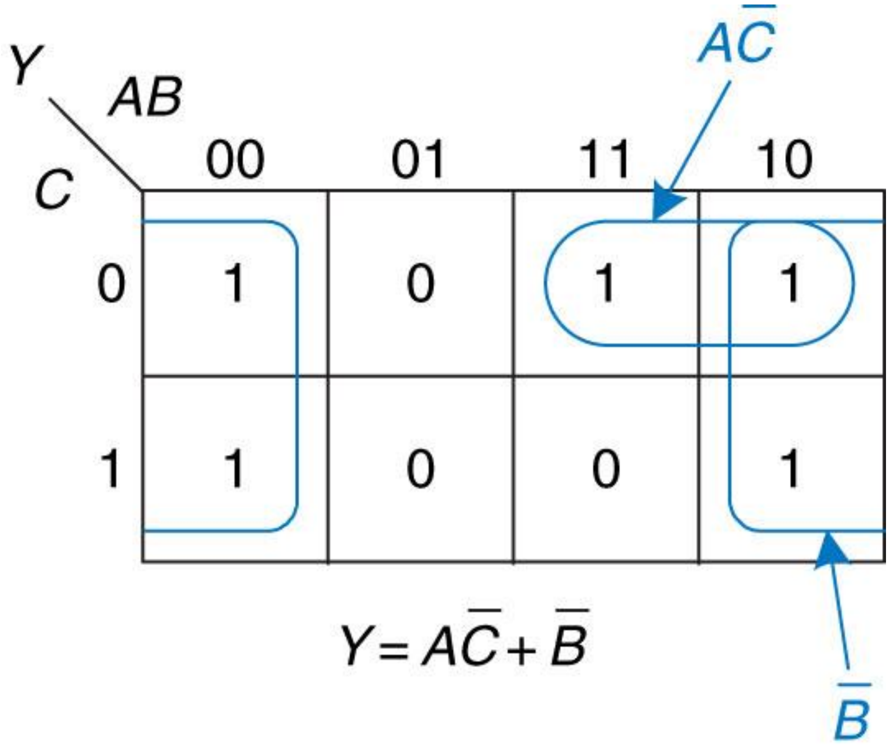
SoP form is:

$$Y = \bar{A}\,B\bar{C} + \bar{A}BC + AB\bar{C}$$

K-map gives:

$$Y = \bar{A}\,B + B\bar{C}$$

Example



SoP: $Y = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + AB\overline{C} + A\overline{B}\,\overline{C} + A\overline{B}C$

Using the K-map: Circle 1s.

Note that we can wrap around.

We cannot do a 3-by-1 rectangle.

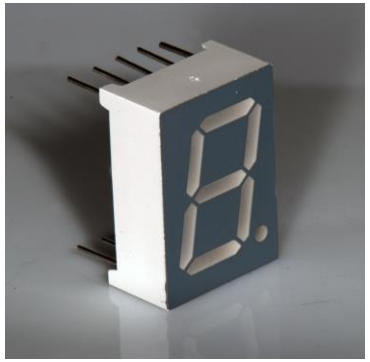
Can still cover the 1s with only 2 rectangles.

Overview of today's lecture

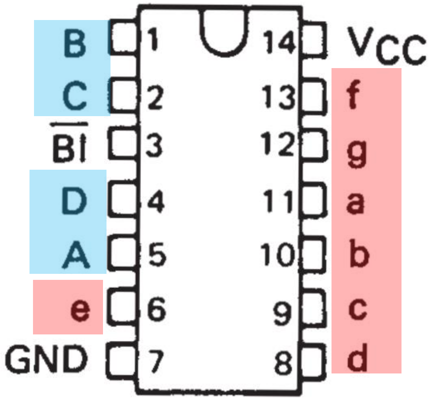
- Karnaugh Maps: a more systematic way to simplify Boolean formulae

- Example: 7-segment display driver

SN7449 BCD to 7-SEGMENT Decoder/Driver



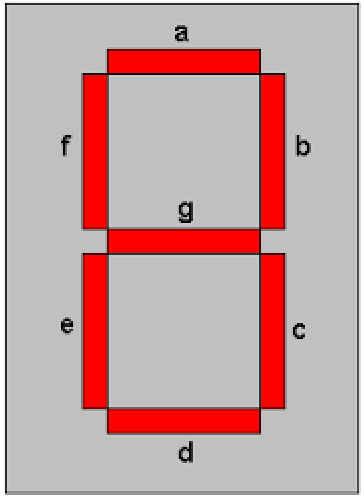
SN54LS49 . . . J OR W PACKAGE
SN74LS49 . . . D OR N PACKAGE
(TOP VIEW)



Binary Inputs				Decoder Outputs							7 Segment Display Outputs
D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	1	0	1	1	9

D_3, D_2, D_1, D_0

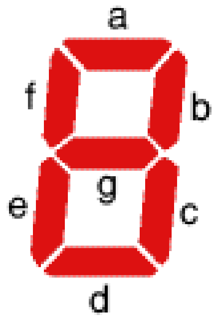
Quickgrid



SEGMENT IDENITIFICATION

Durham University

7-segment display driver



4 inputs D_3, D_2, D_1, D_0 (written $D_{3:0}$).

7 outputs.

Input represents 4-bit binary number.

Output should show corresponding decimal.

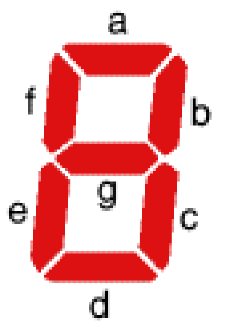
S_a

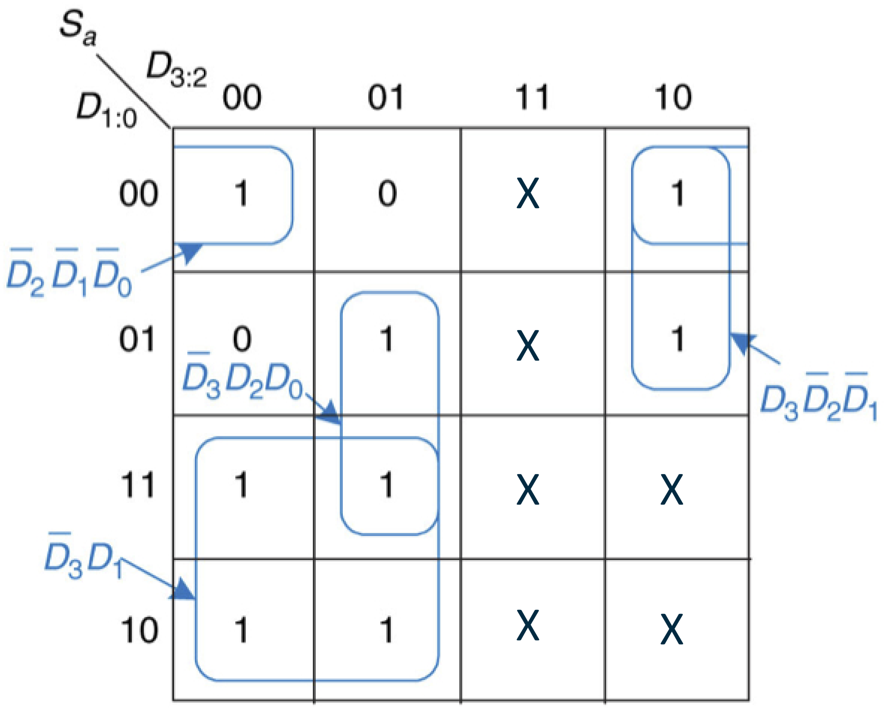
$D_{1:0}$ \ $D_{3:2}$	00	01	11	10
00	1	0	0	1
01	0	1	0	1
11	1	1	0	0
10	1	1	0	0

D_3	D_2	D_1	D_0	S_a
0	0	0	0	**1**
0	0	0	1	**0**
0	0	1	0	**1**
0	0	1	1	**1**
0	1	0	0	**0**
0	1	0	1	**1**
0	1	1	0	**1**
0	1	1	1	**1**
1	0	0	0	**1**
1	0	0	1	**1**

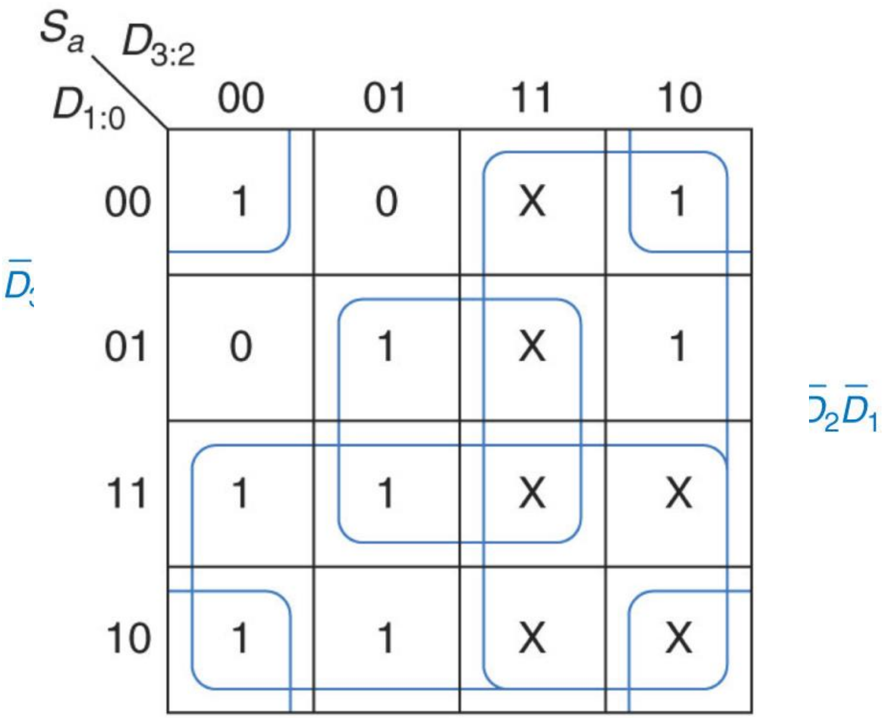
Other inputs' output not specified

7-segment display driver





$$S_a = \bar{D}_3 D_1 + \bar{D}_3 D_2 D_0 + D_3\bar{D}_2\bar{D}_1 + \bar{D}_2\bar{D}_1\bar{D}_0$$

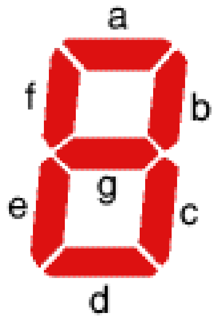


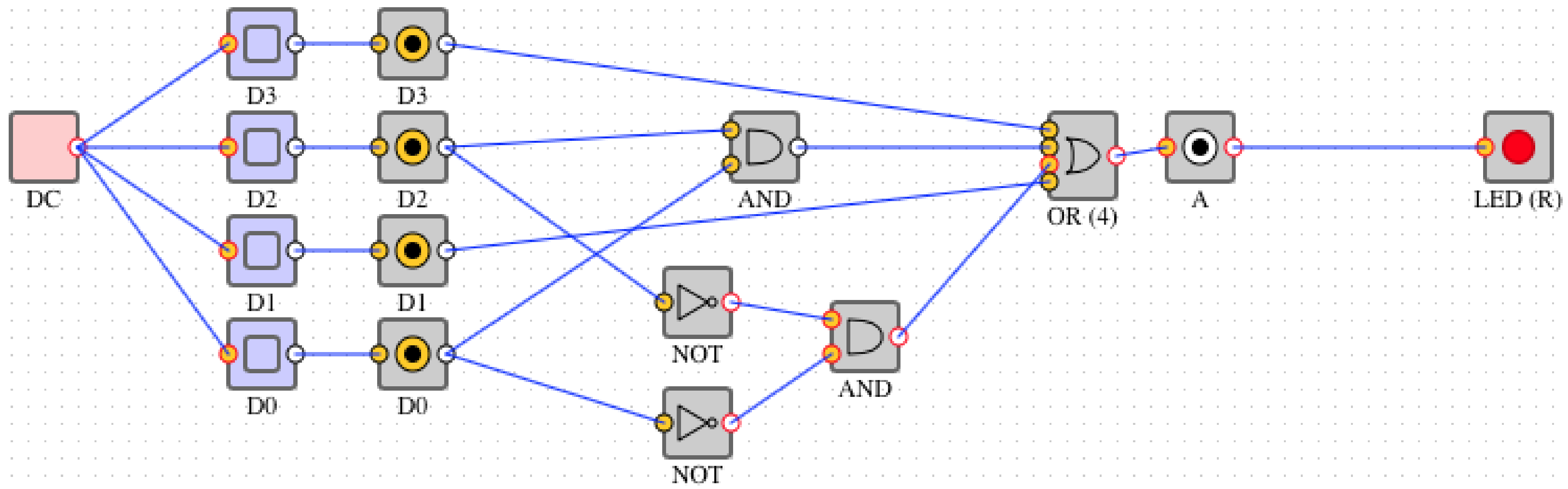
$$S_a = D_3 + D_2 D_0 + \bar{D}_2\bar{D}_0 + D_1$$

What about the unspecified inputs? Have so far assumed they are 0s.

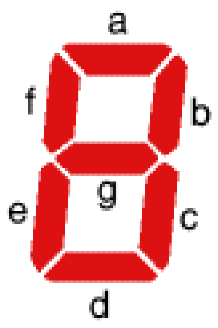
We could equally output 1s if it helped us reduce circuitry!

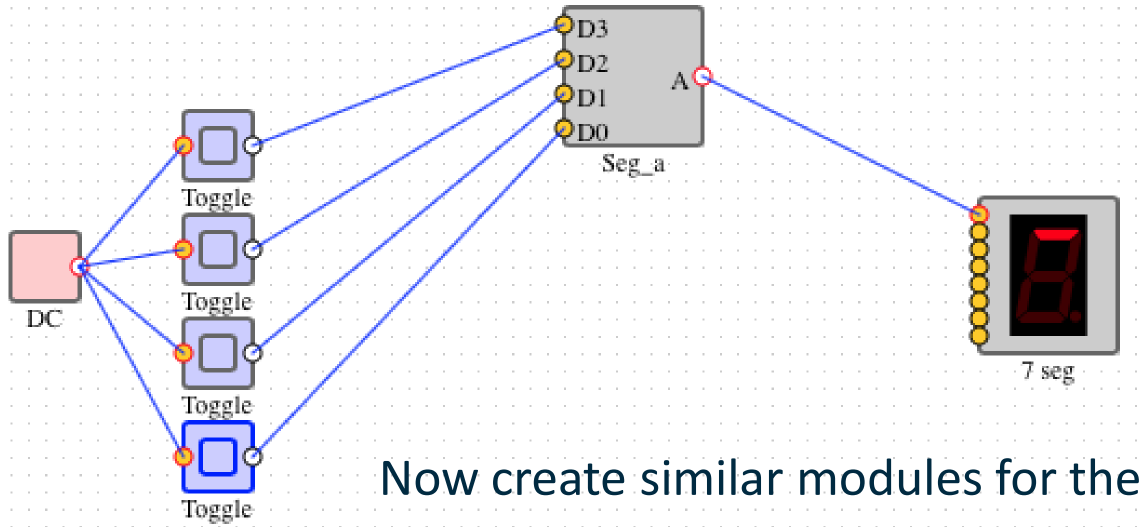
7-segment display driver



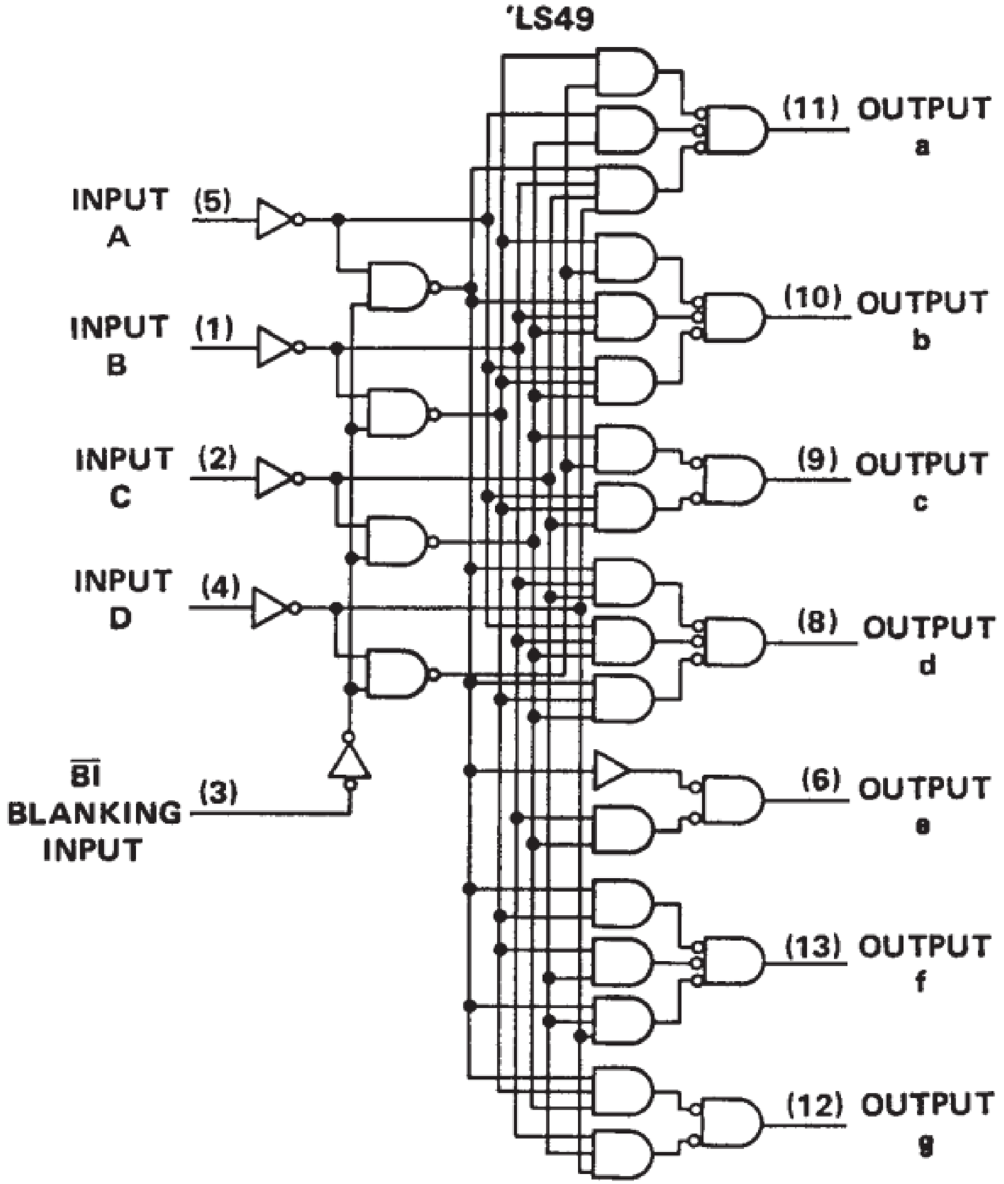


7-segment display driver





Now create similar modules for the other Segments of the display.


'LS49
INPUT A (5)
INPUT B (1)
INPUT C (2)
INPUT D (4)
BI BLANKING INPUT (3)
(11) OUTPUT a
(10) OUTPUT b
(9) OUTPUT c
(8) OUTPUT d
(6) OUTPUT e
(13) OUTPUT f
(12) OUTPUT g

Note: Karnaugh Maps for PoS minimisation

1. Exact same creation of K-map
2. Circle the **zeros** in the map using the same rules:
   - you group 0s **to produce minimum sum terms** instead of grouping 1s to produce minimum product terms
   - Each cell corresponds to a sum term that is false in that cell

$A + \bar{B} + C$

$\bar{A} + B$

Y: C \ AB	00	01	11	10
0	1	0	0	0
1	1	1	1	0

$C + \bar{B}$

Resulting PoS:

$$(C + \bar{B}\ ) \cdot (\bar{A} + B\ )$$

Summary

- Karnaugh Maps: a more systematic way to simplify Boolean formulae

- Example: 7-segment display driver