Digital Electronics

Karnaugh Maps

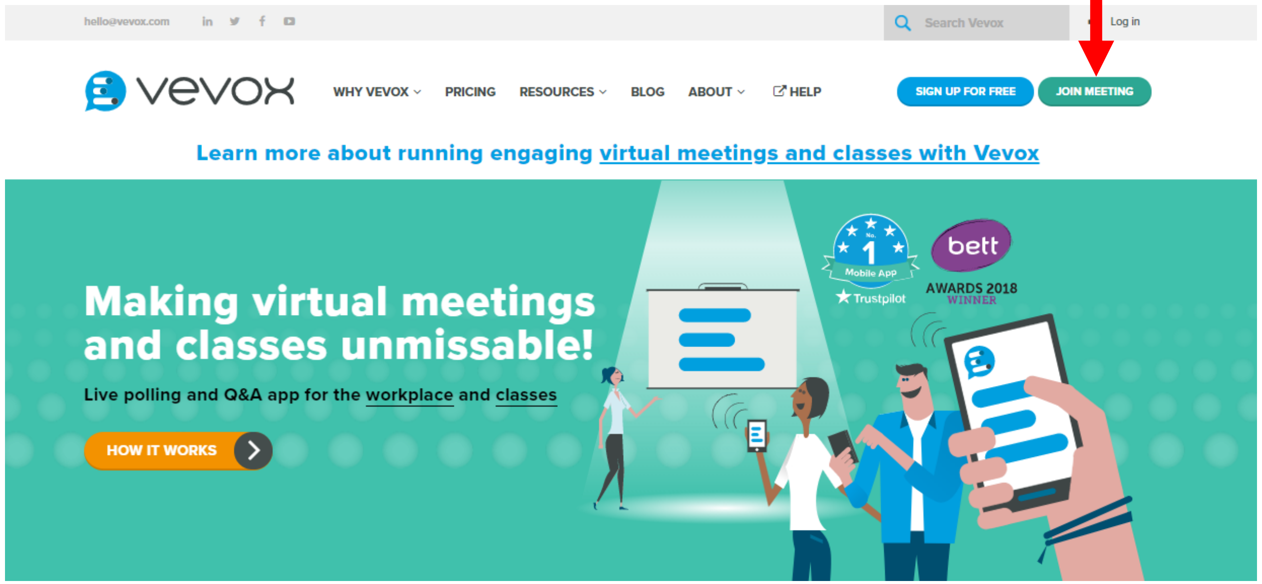
Dr. Eleni Akrida
eleni.akrida@durham.ac.uk

Overview of today’s lecture

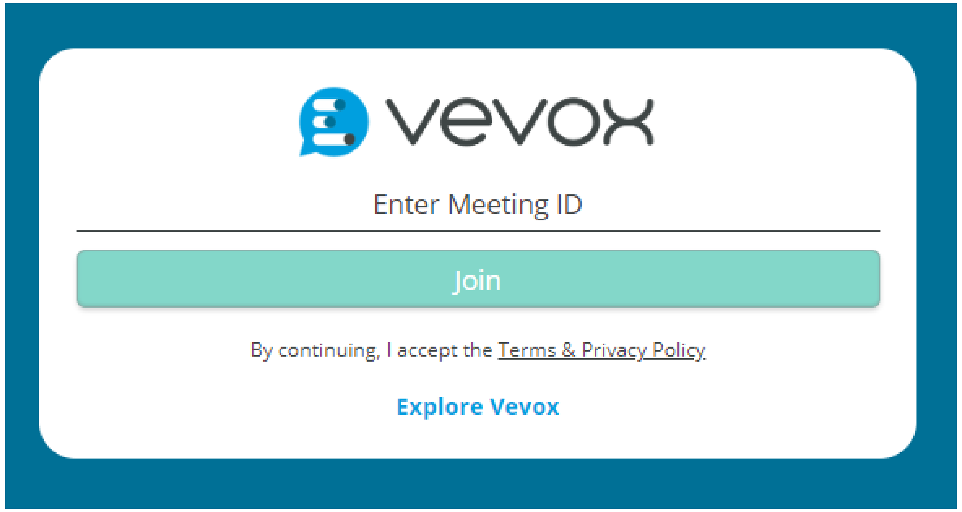
- Karnaugh Maps
  - a more “automated” way to simplify Boolean formulae

- 7-segment display driver

Vevox



Join: **vevox.app** ID: **140-974-939**



Durham University

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

$P(A+\bar{A}) = P \cdot 1 = P$

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).

Durham University

Simplifying Boolean expressions

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

000 010 110 100

000 100

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

ABC
ABC

000 → 1 010 → 0
001 → 1 011 → 0

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

Reduced: $Y = \bar{A}\bar{B}$

AND → 1

0 0	0
0 1	0
1 0	0
1 1	1

OR → 0

0 0	0
0 1	1
1 0	1
1 1	1

XOR
0
1
1
0

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

$\bar{A}B$

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

$\cancel{A}\,B\bar{C}$

$\cdot$ 010 110

SoP form is:

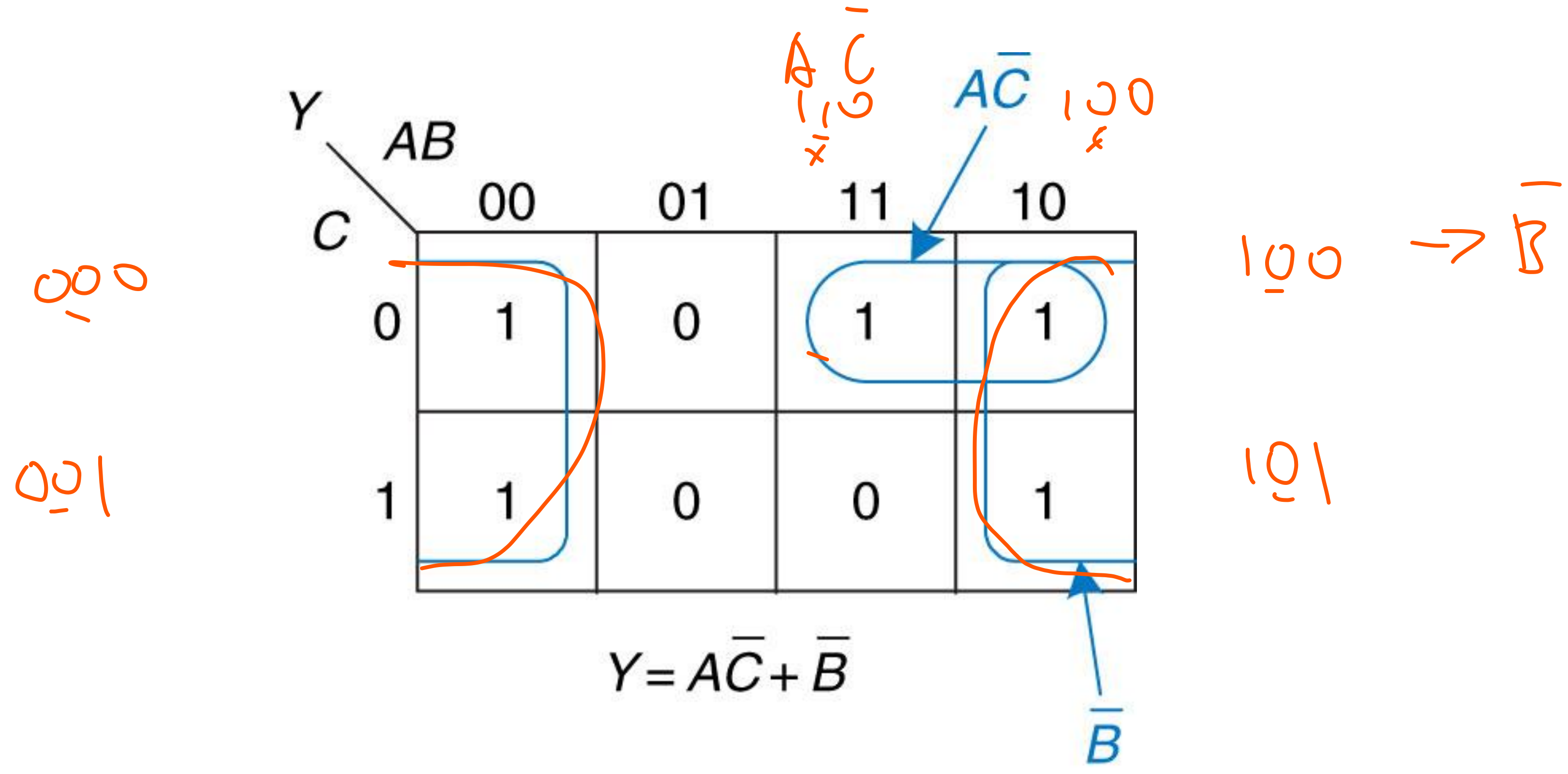
$$Y = \bar{A}\,B\bar{C} + \bar{A}BC + AB\bar{C}$$

K-map gives:

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

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

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.

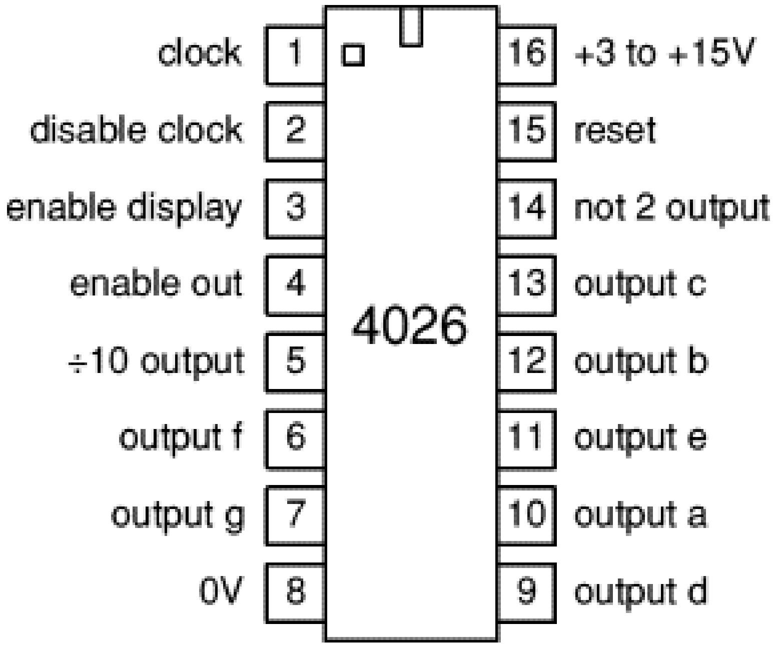
Note that we can wrap around.

We cannot do a 3-by-1 rectangle.

Can still cover the 1s with only 2 rectangles.

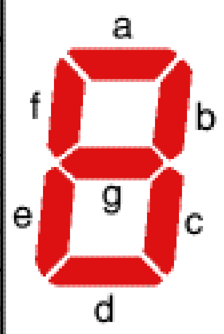
Durham University

4026 decade counter and 7-segment display driver



Outputs from the 4026 counter and display driver IC

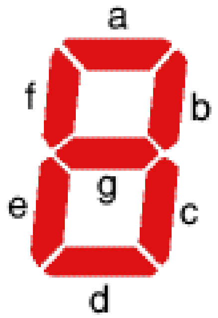
Count	a	b	c	d	e	f	g	h
0	●	●	●	●	●	●		●
1		●	●					●
2	●	●		●	●		●	●
3	●	●	●	●			●	●
4		●	●			●	●	●
5	●		●	●		●	●	
6	●		●	●	●	●	●	
7	●	●	●					
8	●	●	●	●	●	●	●	
9	●	●	●	●		●	●	



7-segment display

● = segment on. h is used to drive other counters.

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.

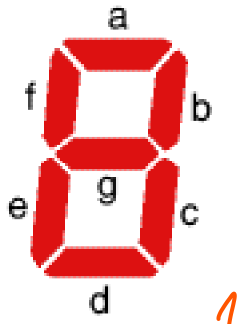
S_a: $D_{1:0}$ \ $D_{3:2}$	00	01	11	10
00	1	0	$\cancel{0}$	1
01	0	1	$\cancel{0}$	1
11	1	1	$\cancel{0}$	$\cancel{0}$
10	1	1	$\cancel{0}$	$\cancel{0}$

	D_3	D_2	D_1	D_0	S_a
0	0	0	0	0	**1**
1	0	0	0	1	**0**
2	0	0	1	0	**1**
3	0	0	1	1	**1**
4	0	1	0	0	**0**
5	0	1	0	1	**1**
6	0	1	1	0	**1**
7	0	1	1	1	**1**
8	1	0	0	0	**1**
9	1	0	0	1	**1**

Other inputs – output not specified

Durham University

7-segment display driver



1) 0000 , 1000 $\bar{D}_2 \bar{D}_1 \bar{D}_0$

2) 0101 0111 $\bar{D}_3 D_2 D_0$

3) 0011 0111
0010 0110
$\bar{D}_3 D_1$

4) 1000
1001 $D_3 \bar{D}_2 \bar{D}_1$

S_a ($D_{1:0}$ \ $D_{3:2}$)	00	01	11	10
00	1	0	X	1
01	0	1	X	1
11	1	1	X	X
10	1	1	X	X

$\bar{D}_2 \bar{D}_1 \bar{D}_0$

$\bar{D}_3 D_2 D_0$

$D_3 \bar{D}_2 \bar{D}_1$

$\bar{D}_3 D_1$

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

D_3

S_a ($D_{1:0}$ \ $D_{3:2}$)	00	01	11	10
00	1	0	X	1
01	0	1	X	1
11	1	1	X	X
10	1	1	X	X

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

1 four corners.
1000 0000
1010 0010
$\bar{D}_2 \bar{D}_0$

2) 0101 1101
0111 1111
$D_2 D_0$

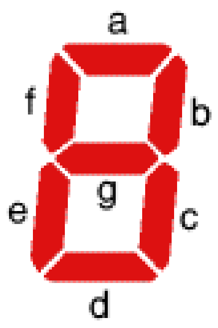
3) ~~1100 1000~~
D_3

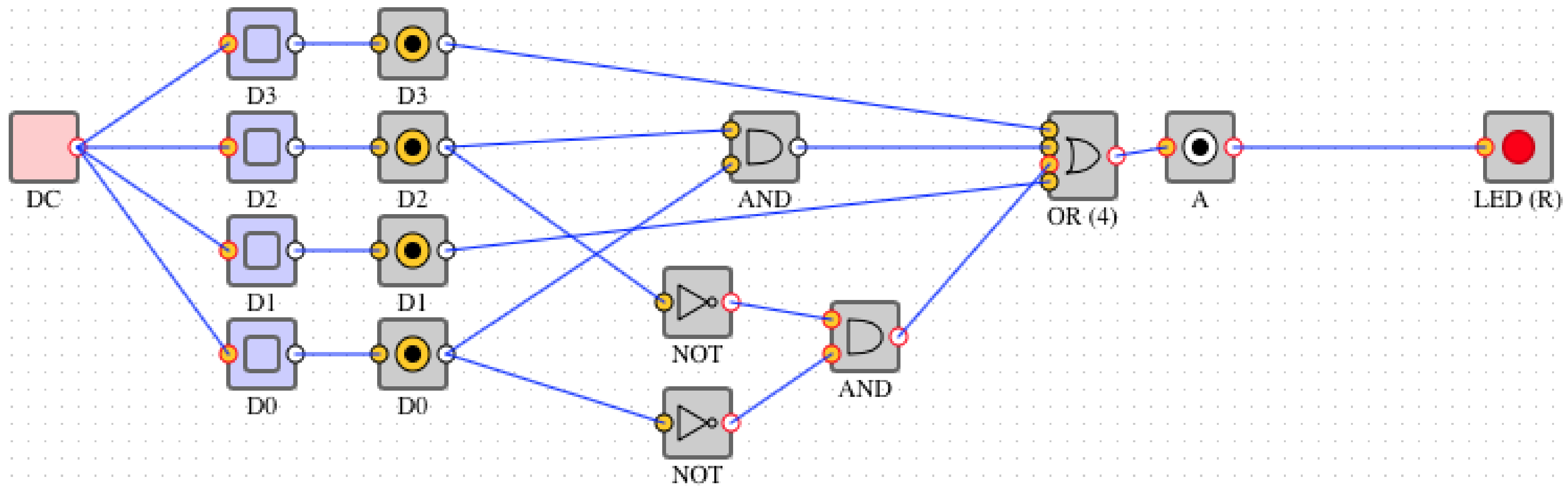
4) D_1

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

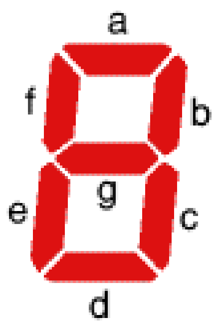
We could equally output 1s if it helped us reduce circuitry!

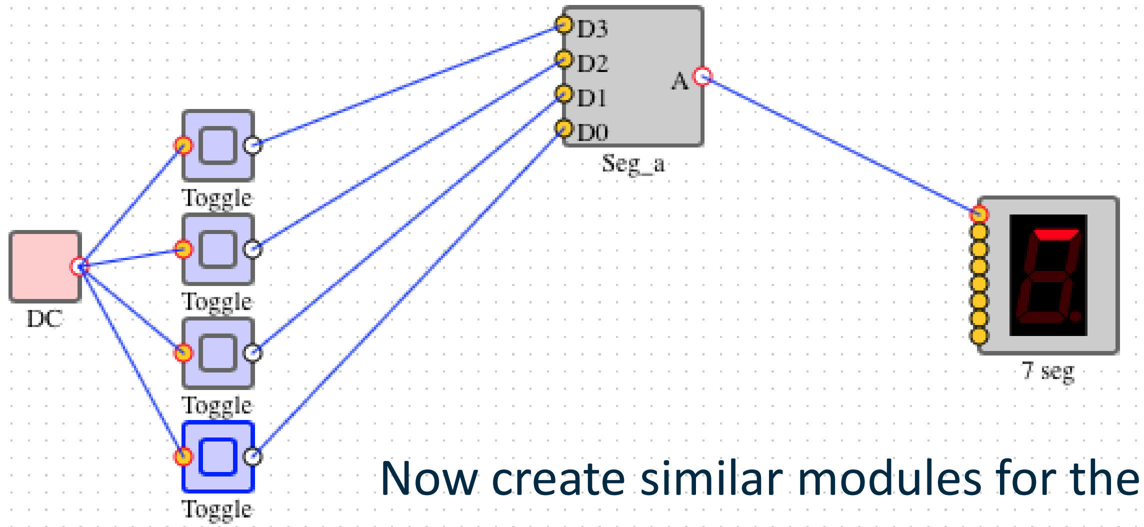
7-segment display driver





7-segment display driver





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

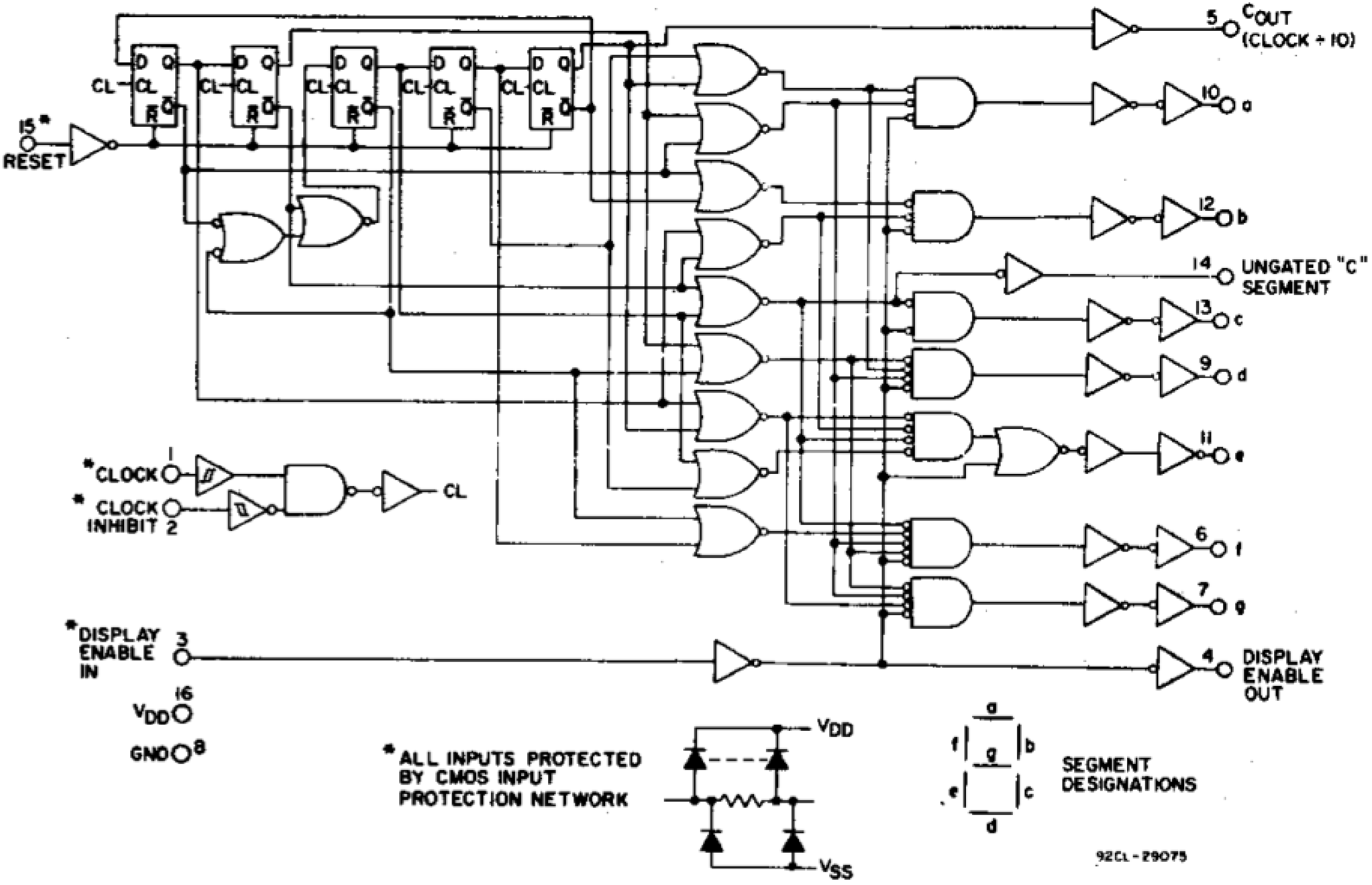


Fig. 1 – CD4026B logic diagram.