BLG231E Digital Circuits

RECITATION-02

Istanbul Technical University

Dec 11, 2020

Exercise 01: Quine-McCluskey Method

Find all prime implicants of the function F using the Quine-McCluskey method.

- *F (a, b, c, d) =* Σ *(0, 1, 3, 7, 8, 9, 11, 15)*

	ABCD
0	0000
1	0001
	0010
	0011
	0100
	0101
	0110
	0111
	1000
	1001
	1010
	1011
	1100
	1101
	1110
	1111

Step 1

Group	Minterm	A	B	C	D	
		Binary Represen.				
0	m_0	0	0	0	0	✓
1	m_1	0	0	0	1	✓
	m_8	1	0	0	0	✓
2	m_3	0	0	1	1	✓
	m_9	1	0	0	1	✓
3	m_7	0	1	1	1	✓
	m_{11}	1	0	1	1	✓
4	m_{15}	1	1	1	1	✓

Step 2

Group	Matched Pairs	A	B	C	D
0	m_0-m_1	0	0	0	–
	m_0-m_8	–	0	0	0
1	m_1-m_3	0	0	–	1
	m_1-m_9	–	0	0	1
	m_8-m_9	1	0	0	–
2	m_3-m_7	0	–	1	1
	m_3-m_{11}	–	0	1	1
	m_9-m_{11}	1	0	–	1
3	m_7-m_{15}	–	1	1	1
	$m_{11}-m_{15}$	1	–	1	1

Exercise 01: Quine-McCluskey Method

Find all prime implicants of the function F using the Quine-McCluskey method.

- *F (a, b, c, d) =* Σ *(0, 1, 3, 7, 8, 9, 11, 15)*

Step 2

Group	Matched Pairs	A	B	C	D
0	$m_0 - m_1$	0	0	0	–
	$m_0 - m_8$	–	0	0	0
1	$m_1 - m_3$	0	0	–	1
	$m_1 - m_9$	–	0	0	1
	$m_8 - m_9$	1	0	0	–
2	$m_3 - m_7$	0	–	1	1
	$m_3 - m_{11}$	–	0	1	1
	$m_9 - m_{11}$	1	0	–	1
3	$m_7 - m_{15}$	–	1	1	1
	$m_{11} - m_{15}$	1	–	1	1

Group	Matched Pairs	A	B	C	D	
0	$m_0 - m_1 - m_8 - m_9$	–	0	0	–	$\bar{B}\bar{C}$
	$m_0 - m_8 - m_1 - m_9$	–	0	0	–	
1	$m_1 - m_3 - m_9 - m_{11}$	–	0	–	0	$\bar{B}D$
	$m_1 - m_9 - m_3 - m_{11}$	–	0	–	0	
2	$m_3 - m_7 - m_{11} - m_{15}$	–	–	1	1	CD
	$m_3 - m_{11} - m_7 - m_{15}$	–	–	1	1	

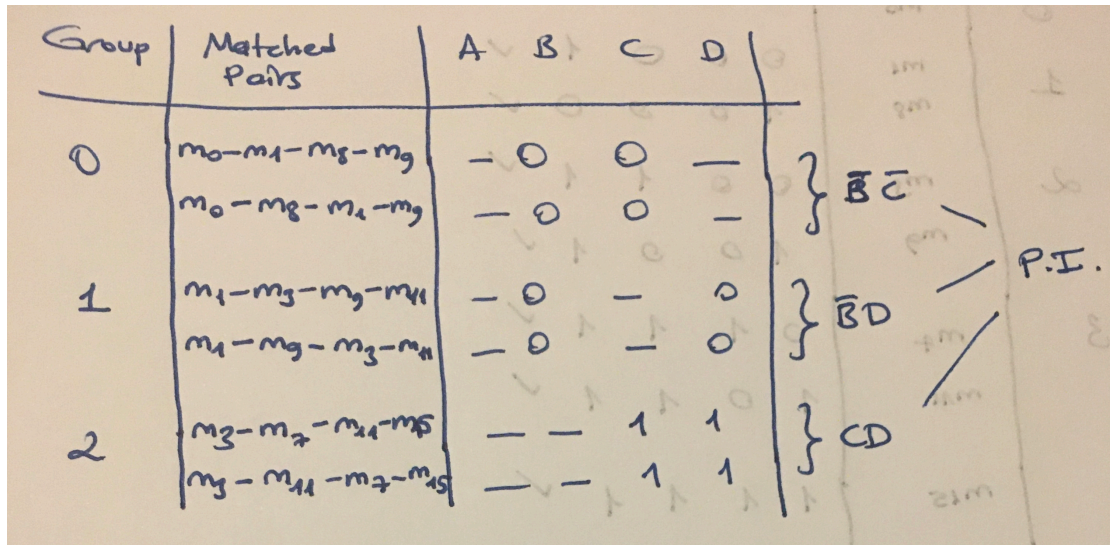
P.I.

Step 3

Exercise 01: Quine-McCluskey Method

Find all prime implicants of the function F using the Quine-McCluskey method.

- *F (a, b, c, d) =* Σ *(0, 1, 3, 7, 8, 9, 11, 15)*

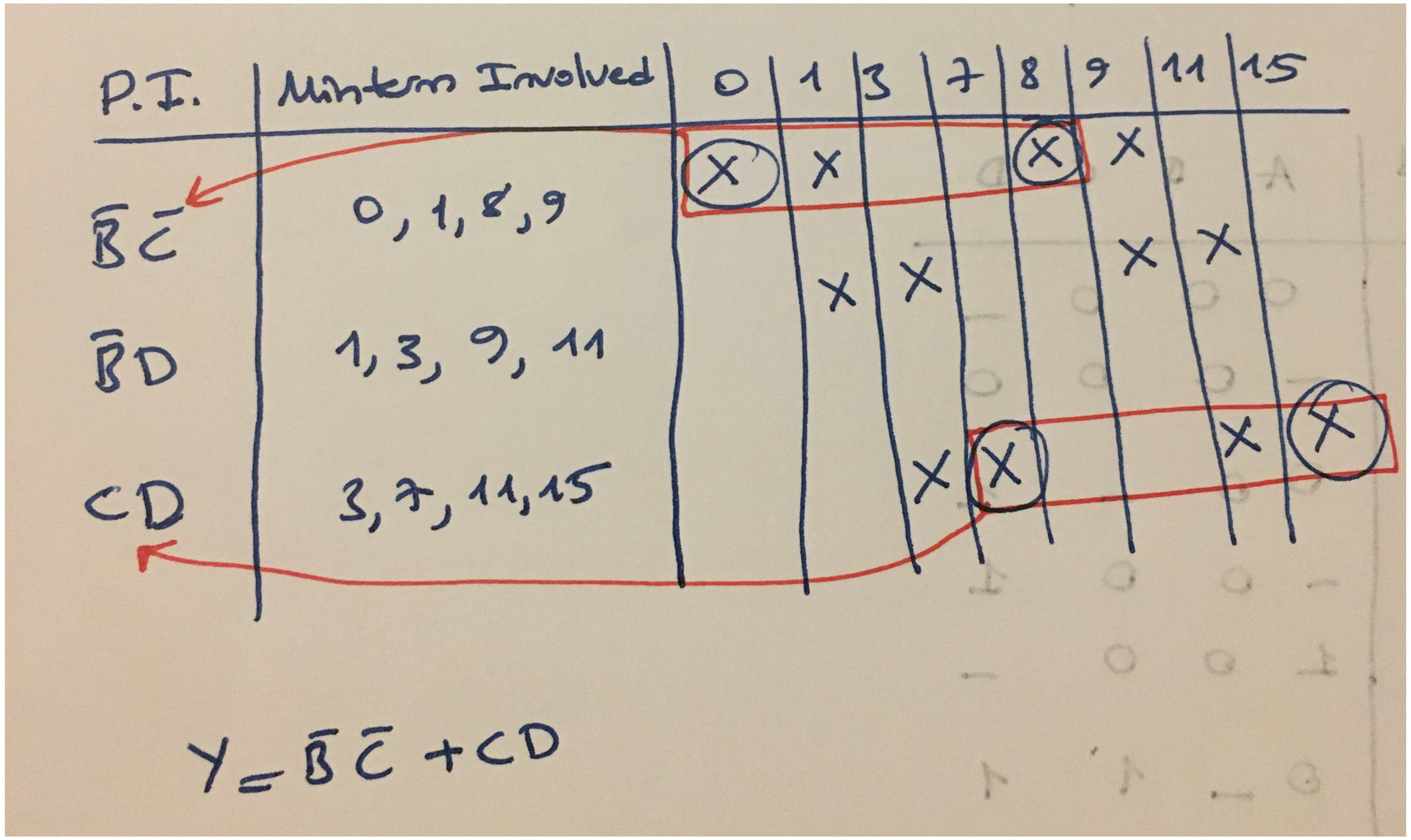


Group	Matched Pairs	A	B	C	D	
0	$m_0-m_1-m_8-m_9$	–	0	0	–	$\bar{B}\bar{C}$
	$m_0-m_8-m_1-m_9$	–	0	0	–	
1	$m_1-m_3-m_9-m_{11}$	–	0	–	0	$\bar{B}D$
	$m_1-m_9-m_3-m_{11}$	–	0	–	0	
2	$m_3-m_7-m_{11}-m_{15}$	–	–	1	1	CD
	$m_3-m_{11}-m_7-m_{15}$	–	–	1	1	

P.I.

Step 3

Step 4: PI Chart



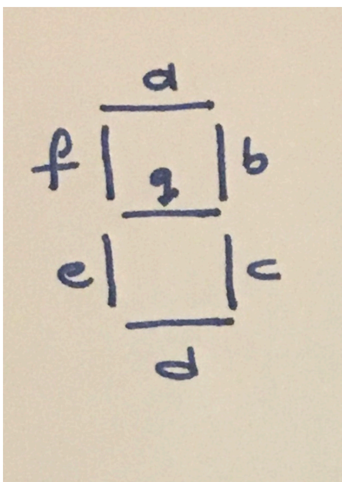
P.I.	Minterm Involved	0	1	3	7	8	9	11	15
$\bar{B}\bar{C}$	0, 1, 8, 9	X	X			X	X		
$\bar{B}D$	1, 3, 9, 11		X	X			X	X	
CD	3, 7, 11, 15			X	X			X	X

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

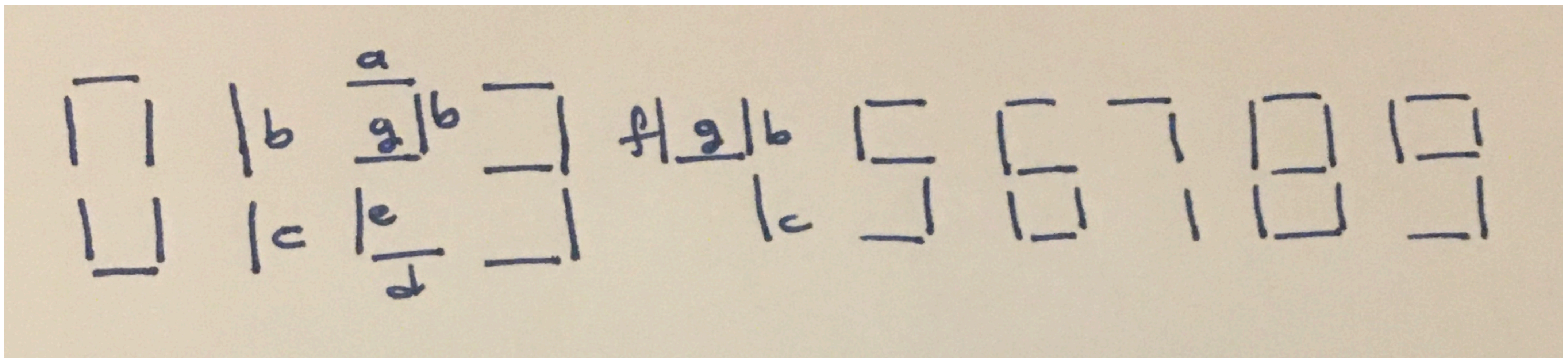
Exercise 02: Seven-Segment Display

An ABCD-to-seven-segment decoder is combinational circuit that converts a decimal digit in BCD to an appopriate code for the selection of segments in an indicator used to display the decimal digit in a familiar form.

- The seven outputs of the decoder `(a, b, c, d, e, f, g)` select the corresponding segments in the display. The numerical display chosen to represent the decimal digit is shown below.



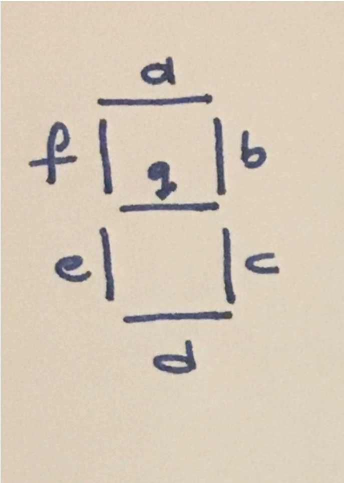
Segment designation



Numerical designation for display

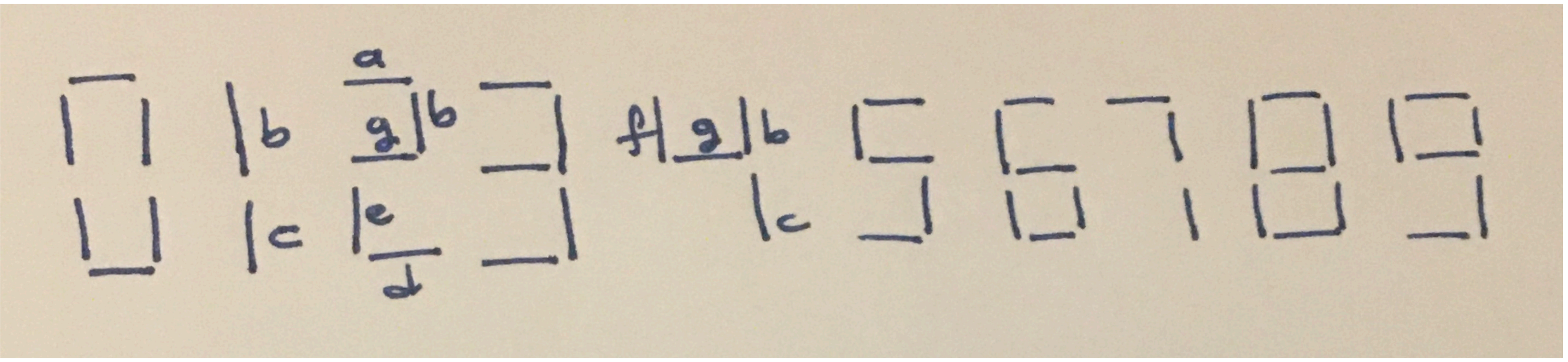
- Using a truth table and Karnaugh maps, design the BCD-to-seven-segment decoder, using a minimum number of gates.
- Invalid combinations such as `(b,f), (a, e)`, will result in a blank display.

- Using a truth table and Karnaugh maps, design the BCD-to-seven-segment decoder, using a minimum number of gates.



Segment designation

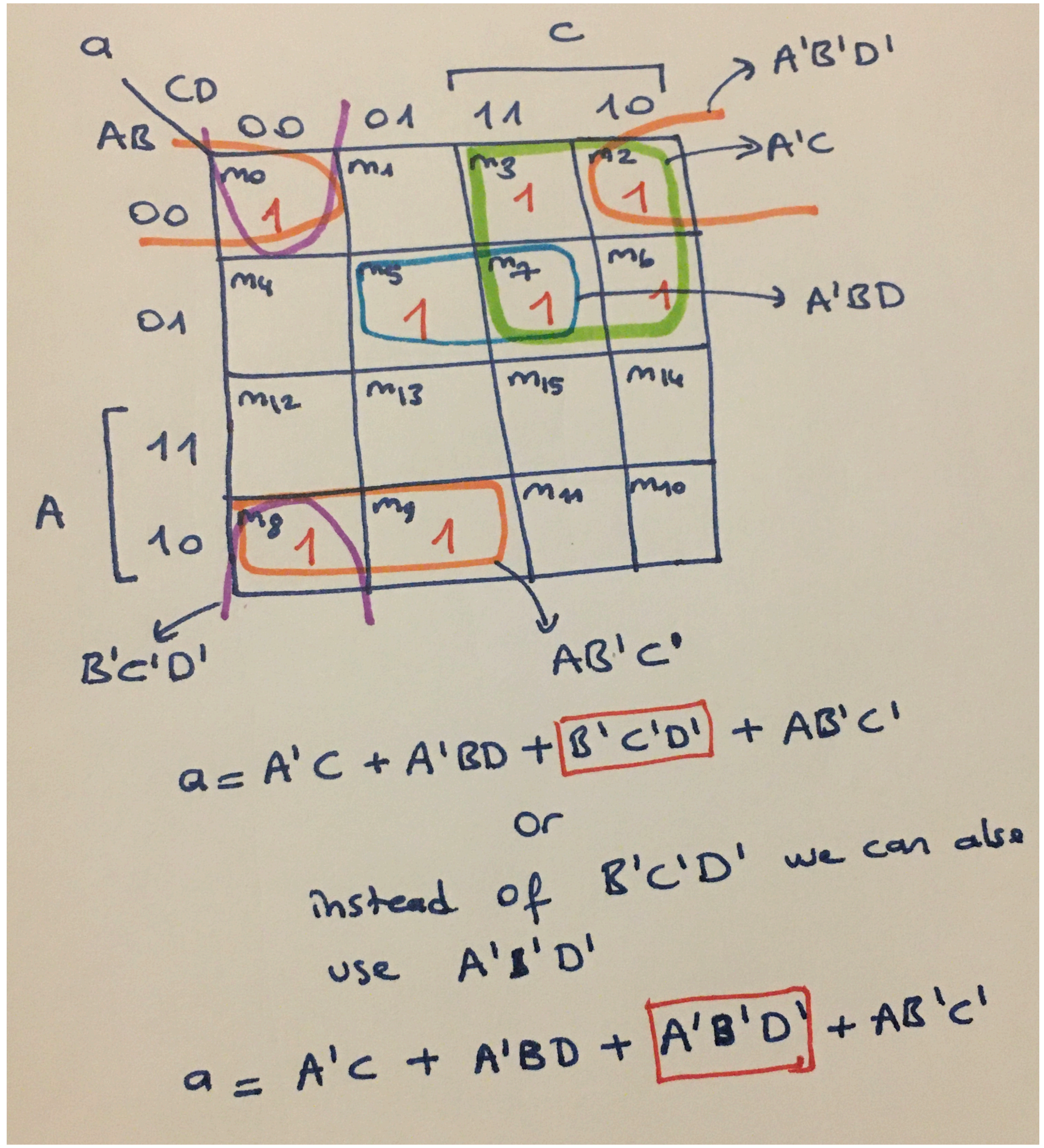
ABCD	a	b	c	d	e	f	g
0000	1	1	1	1	1	1	0
0001	0	1	1	0	0	0	0
0010	1	1	0	1	1	0	1
0011	1	1	1	1	0	0	1
0100	0	1	1	0	0	1	1
0101	1	0	1	1	0	1	1
0110	1	0	1	1	1	1	1
0111	1	1	1	0	0	0	0
1000	1	1	1	1	1	1	1
1001	1	1	1	1	0	1	1



Numerical designation for display

- Using a truth table and Karnaugh maps, design the BCD-to-seven-segment decoder, using a minimum number of gates.

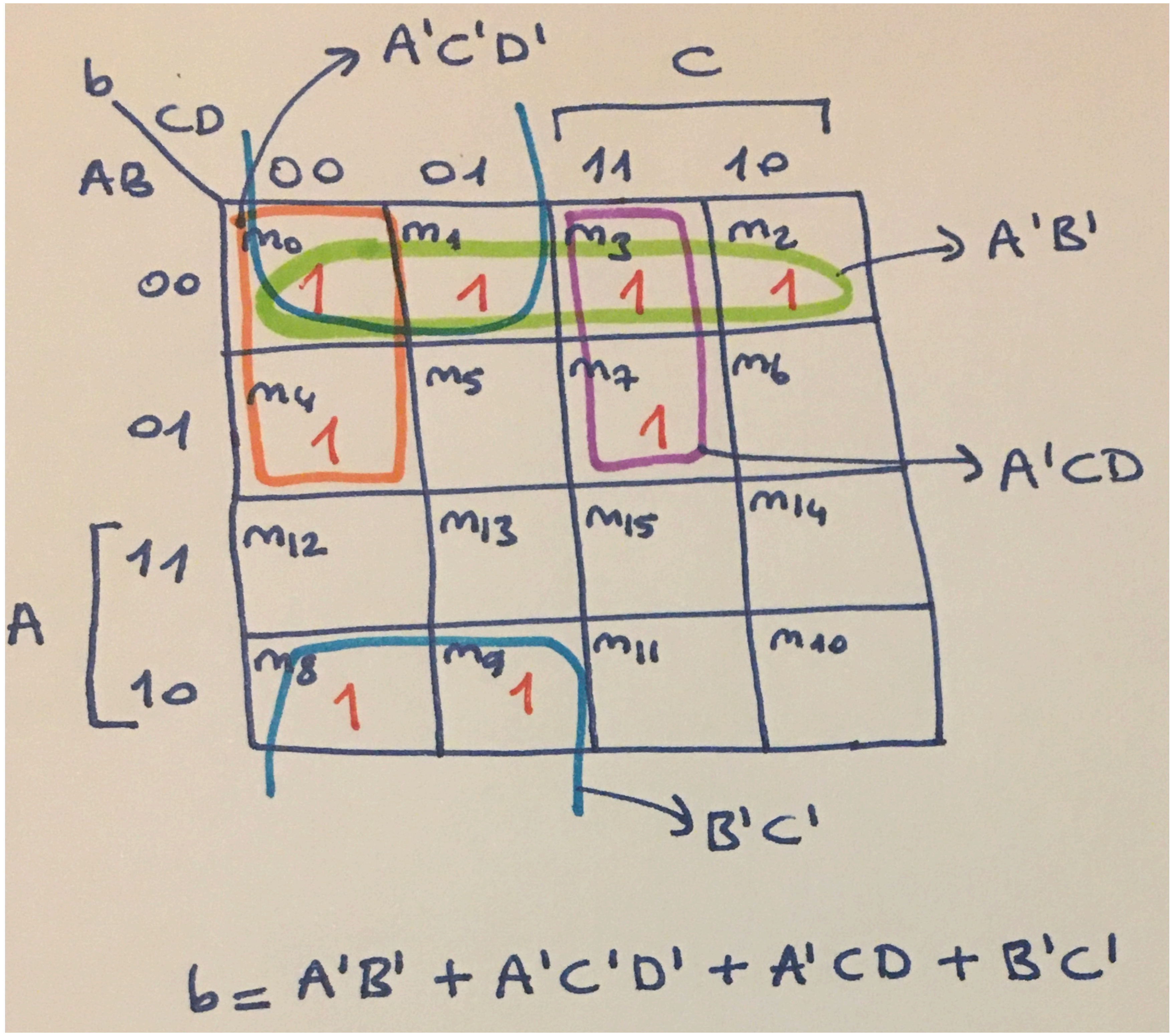
ABCD	a	b	c	d	e	f	g
0000	1	1	1	1	1	1	0
0001	0	1	1	0	0	0	0
0010	1	1	0	1	1	0	1
0011	1	1	1	1	0	0	1
0100	0	1	1	0	0	1	1
0101	1	0	1	1	0	1	1
0110	1	0	1	1	1	1	1
0111	1	1	1	0	0	0	0
1000	1	1	1	1	1	1	1
1001	1	1	1	1	0	1	1



Exercise 02: Seven-Segment Display

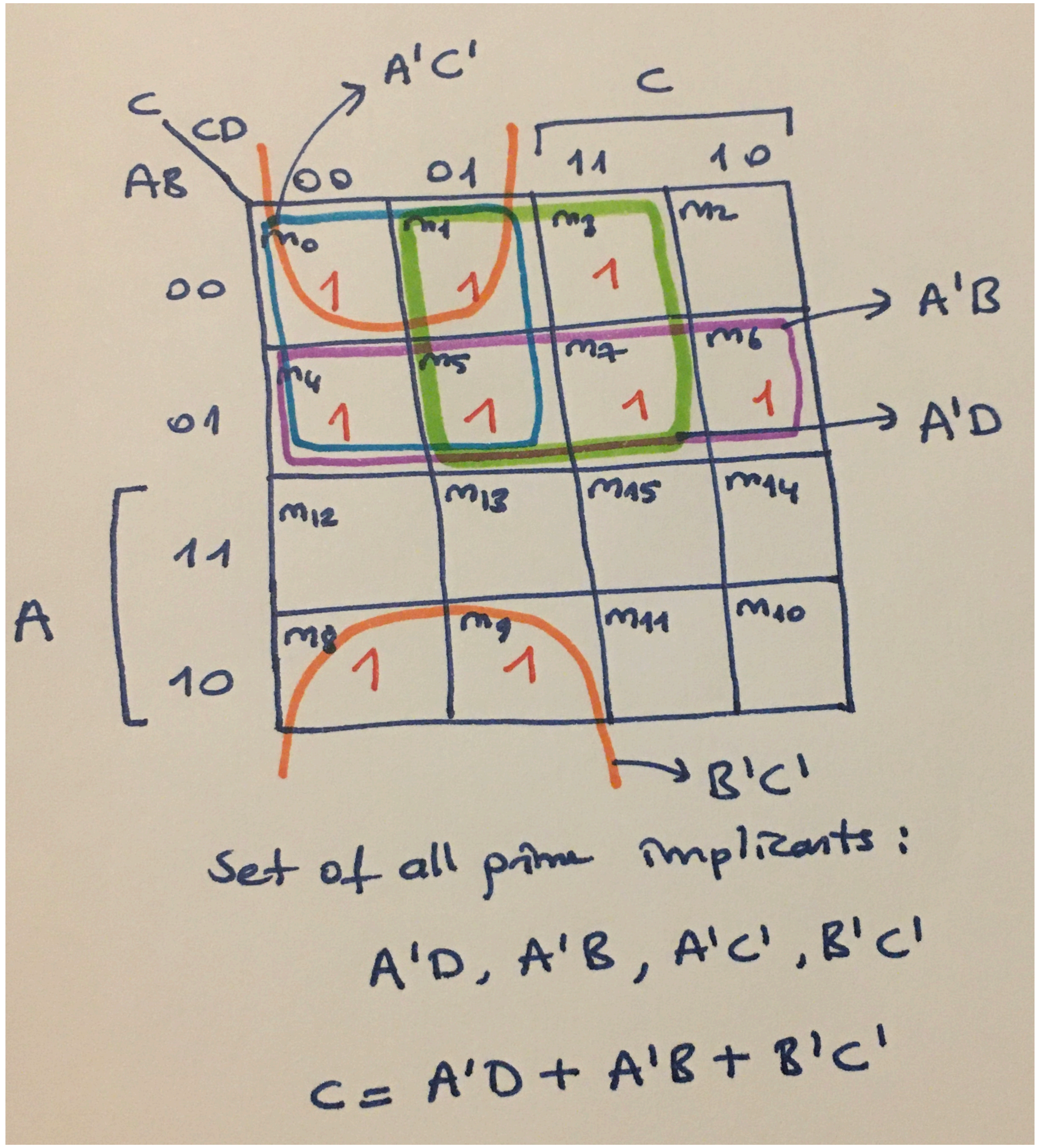
- Using a truth table and Karnaugh maps, design the BCD-to-seven-segment decoder, using a minimum number of gates.

ABCD	a	b	c	d	e	f	g
0000	1	1	1	1	1	1	0
0001	0	1	1	0	0	0	0
0010	1	1	0	1	1	0	1
0011	1	1	1	1	0	0	1
0100	0	1	1	0	0	1	1
0101	1	0	1	1	0	1	1
0110	1	0	1	1	1	1	1
0111	1	1	1	0	0	0	0
1000	1	1	1	1	1	1	1
1001	1	1	1	1	0	1	1



- Using a truth table and Karnaugh maps, design the BCD-to-seven-segment decoder, using a minimum number of gates.

ABCD	a	b	c	d	e	f	g
0000	1	1	1	1	1	1	0
0001	0	1	1	0	0	0	0
0010	1	1	0	1	1	0	1
0011	1	1	1	1	0	0	1
0100	0	1	1	0	0	1	1
0101	1	0	1	1	0	1	1
0110	1	0	1	1	1	1	1
0111	1	1	1	0	0	0	0
1000	1	1	1	1	1	1	1
1001	1	1	1	1	0	1	1



Exercise 02: Seven-Segment Display

- Using a truth table and Karnaugh maps, design the BCD-to-seven-segment decoder, using a minimum number of gates.

ABCD	a	b	c	d	e	f	g
0000	1	1	1	1	1	1	0
0001	0	1	1	0	0	0	0
0010	1	1	0	1	1	0	1
0011	1	1	1	1	0	0	1
0100	0	1	1	0	0	1	1
0101	1	0	1	1	0	1	1
0110	1	0	1	1	1	1	1
0111	1	1	1	0	0	0	0
1000	1	1	1	1	1	1	1
1001	1	1	1	1	0	1	1

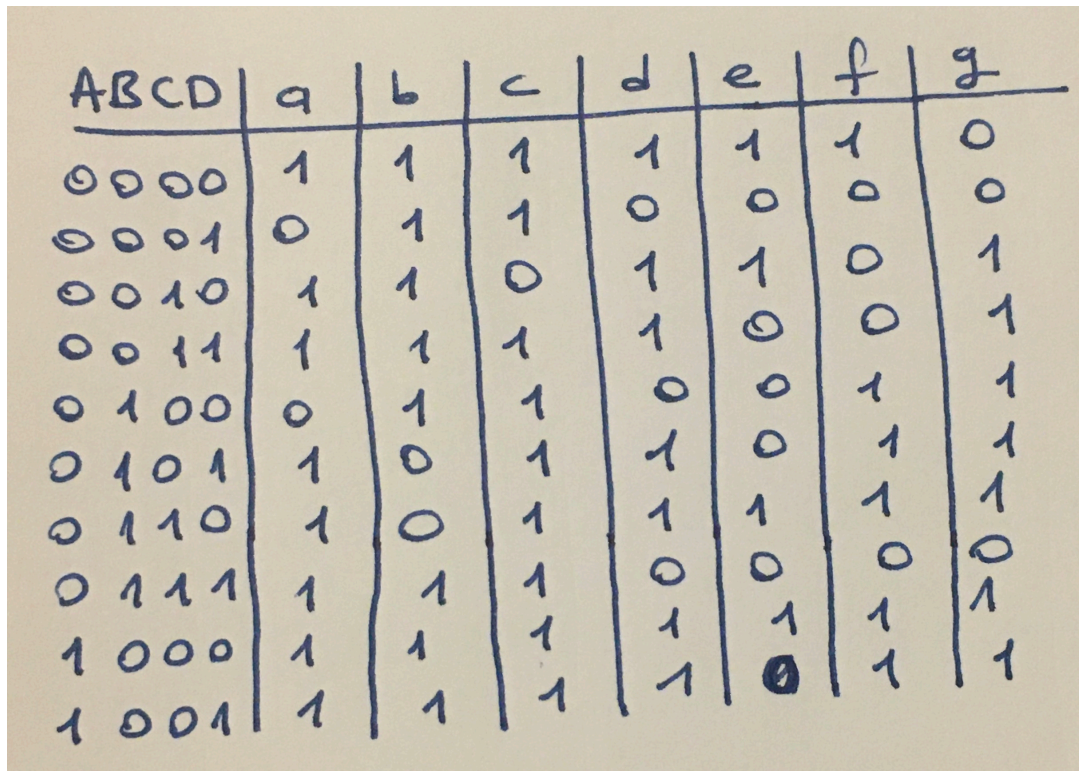
$d = A'CD' + A'B'C + \boxed{B'C'D'} + AB'C' + A'BC'D$

or instead of B'C'D we can also use

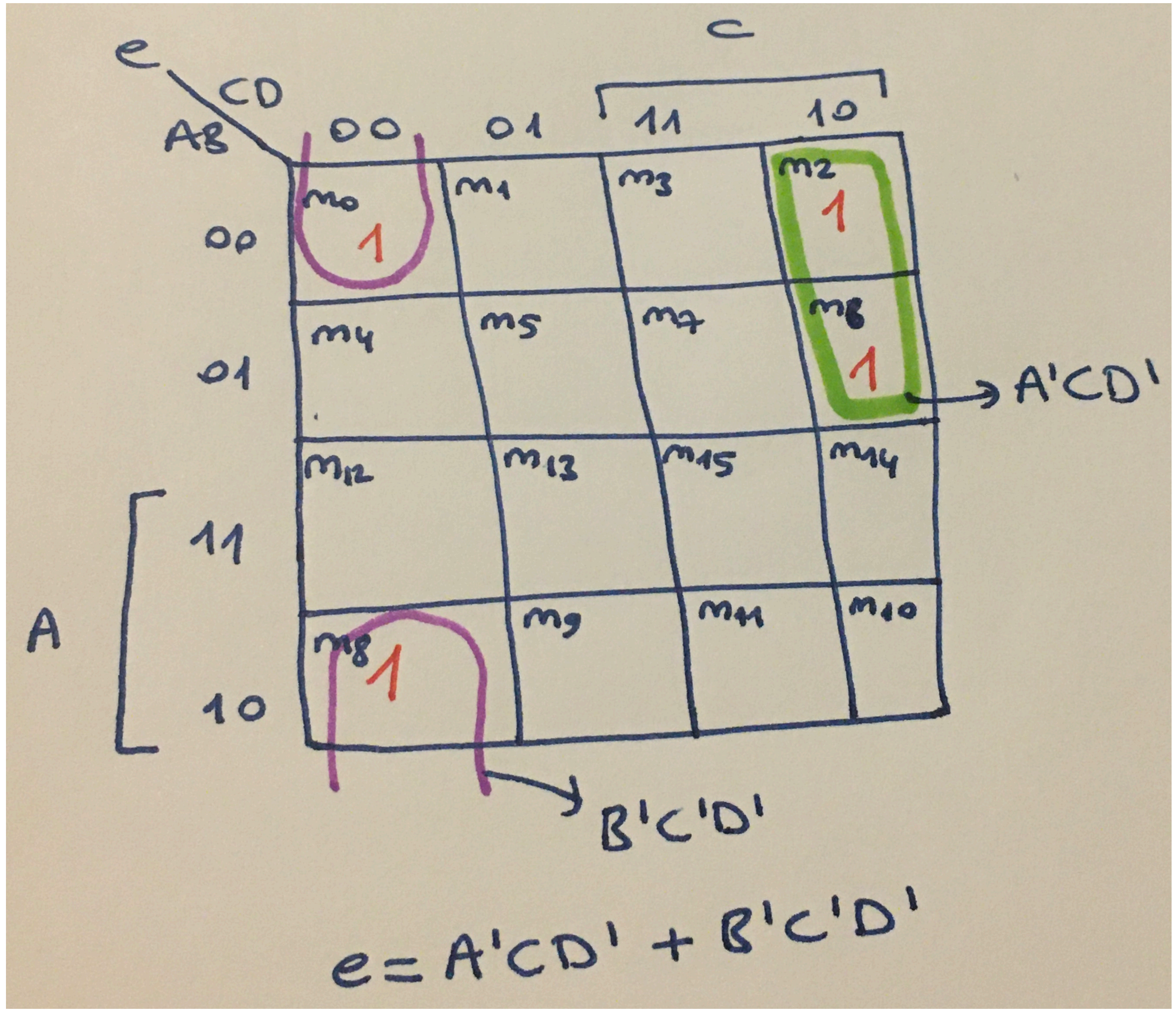
$A'B'D'$

$d = A'CD' + A'B'C + \boxed{A'B'D'} + AB'C' + A'BC'D$

- Using a truth table and Karnaugh maps, design the BCD-to-seven-segment decoder, using a minimum number of gates.



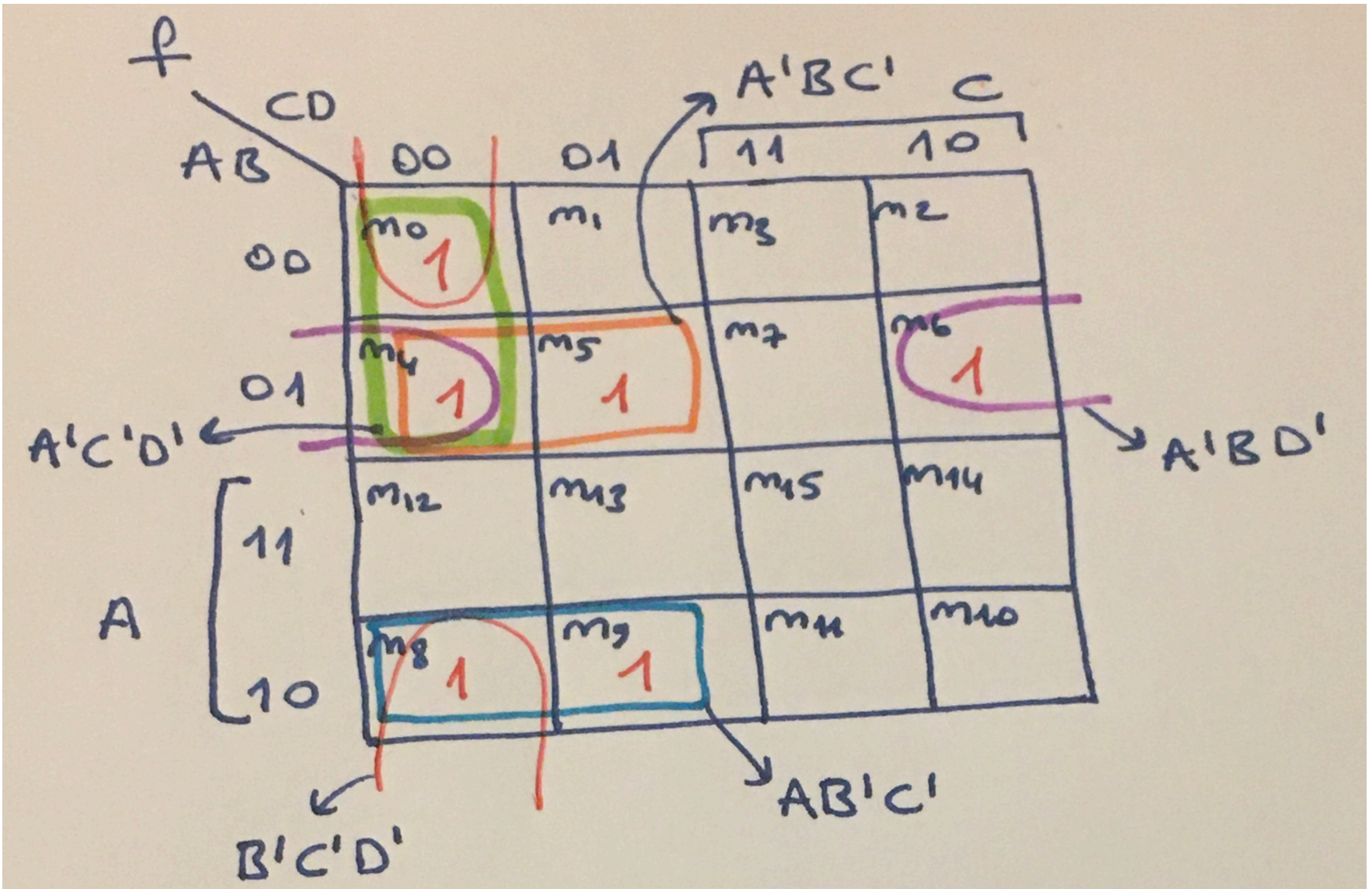
ABCD	a	b	c	d	e	f	g
0000	1	1	1	1	1	1	0
0001	0	1	1	0	0	0	0
0010	1	1	0	1	1	0	1
0011	1	1	1	1	0	0	1
0100	0	1	1	0	0	1	1
0101	1	0	1	1	0	1	1
0110	1	0	1	1	1	1	1
0111	1	1	1	0	0	0	0
1000	1	1	1	1	1	1	1
1001	1	1	1	1	0	1	1



Exercise 02: Seven-Segment Display

- Using a truth table and Karnaugh maps, design the BCD-to-seven-segment decoder, using a minimum number of gates.

ABCD	a	b	c	d	e	f	g
0000	1	1	1	1	1	1	0
0001	0	1	1	0	0	0	0
0010	1	1	0	1	1	0	1
0011	1	1	1	1	0	0	1
0100	0	1	1	0	0	1	1
0101	1	0	1	1	0	1	1
0110	1	0	1	1	1	1	1
0111	1	1	1	0	0	0	0
1000	1	1	1	1	1	1	1
1001	1	1	1	1	0	1	1

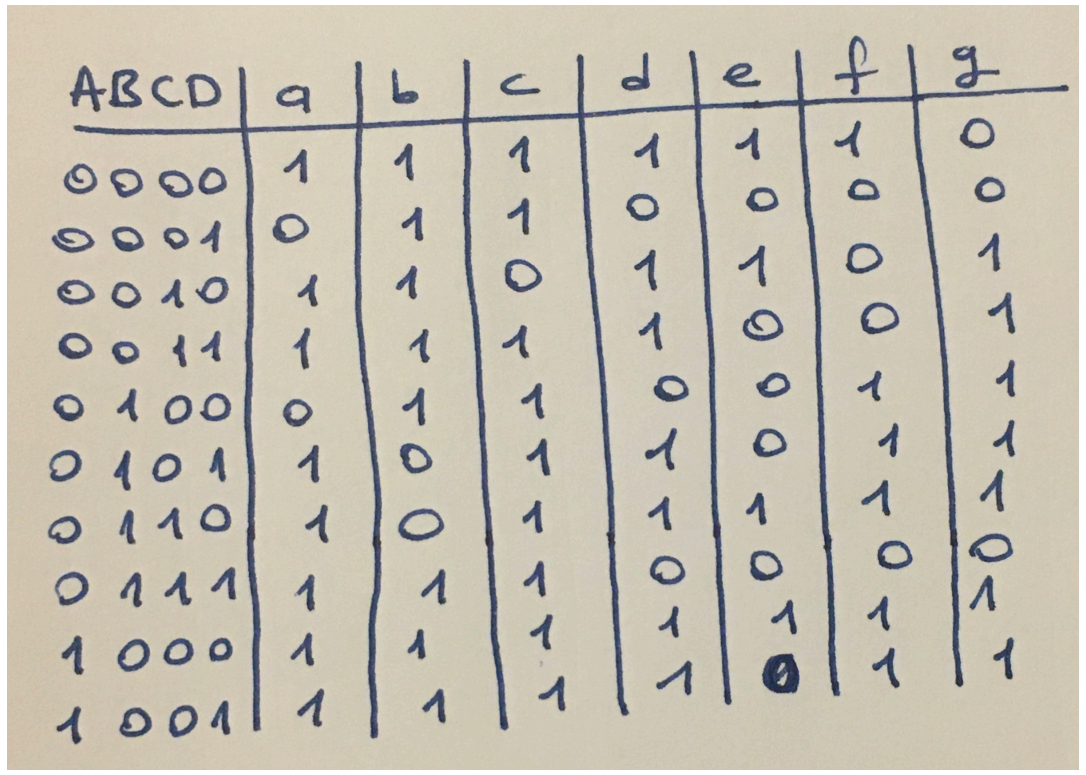


$f = A'BC' + A'C'D' + A'BD' + AB'C'$

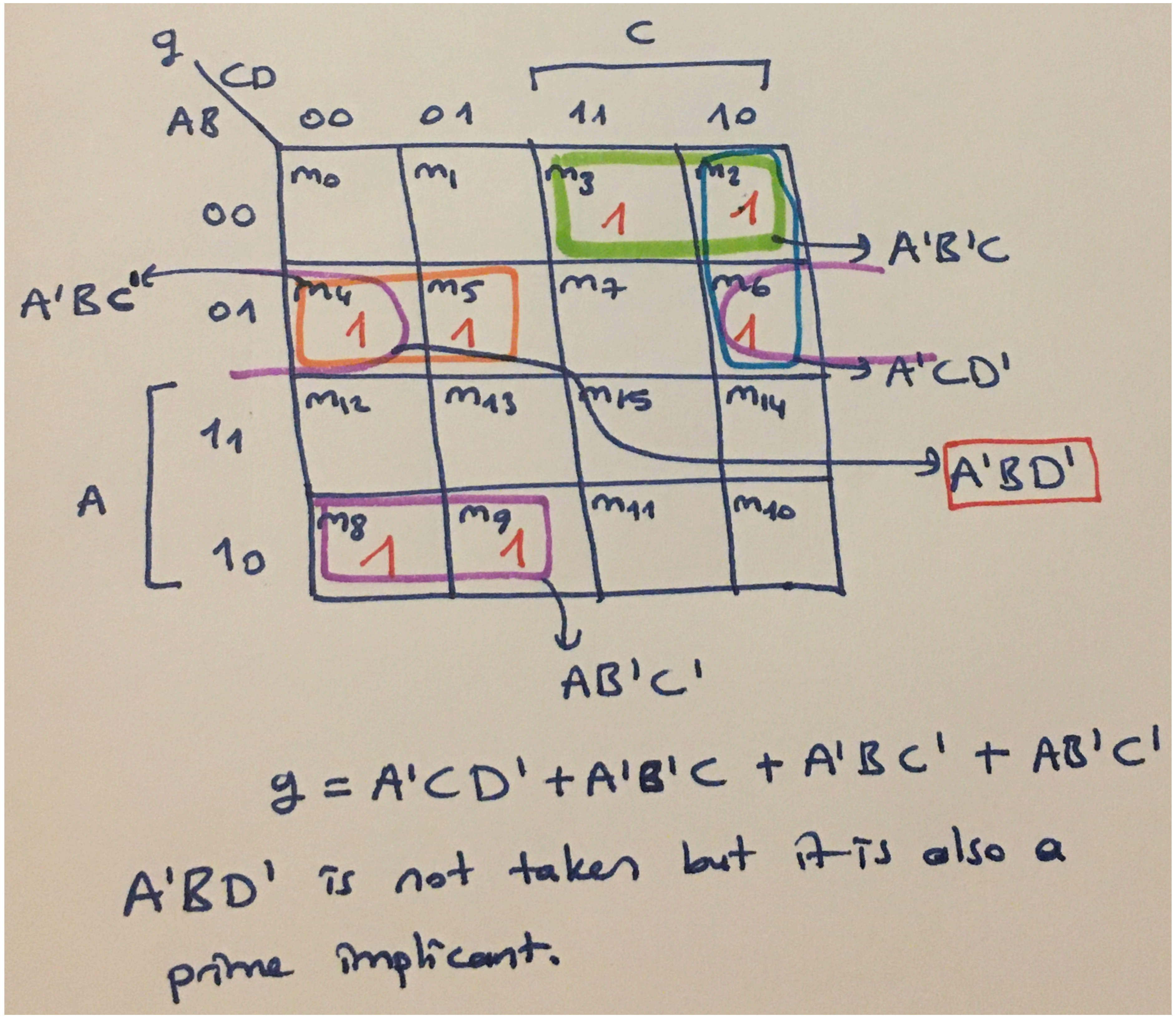
B'C'D' is also a prime implicant but we will not use it as all the others cover every 1 on the K-map.

Exercise 02: Seven-Segment Display

- Using a truth table and Karnaugh maps, design the BCD-to-seven-segment decoder, using a minimum number of gates.



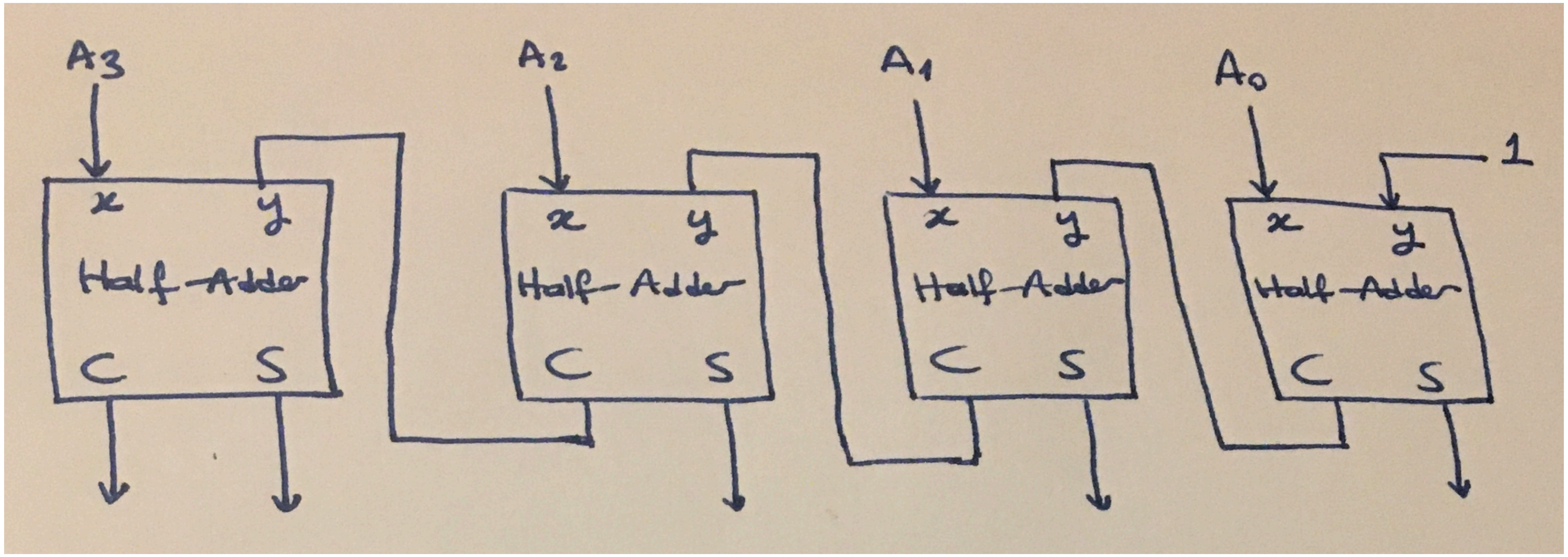
ABCD	a	b	c	d	e	f	g
0000	1	1	1	1	1	1	0
0001	0	1	1	0	0	0	0
0010	1	1	0	1	1	0	1
0011	1	1	1	1	0	0	1
0100	0	1	1	0	0	1	1
0101	1	0	1	1	0	1	1
0110	1	0	1	1	1	1	1
0111	1	1	1	0	0	0	0
1000	1	1	1	1	1	1	1
1001	1	1	1	1	0	1	1



Exercise 03: Half-Adder

Using four half-adders:

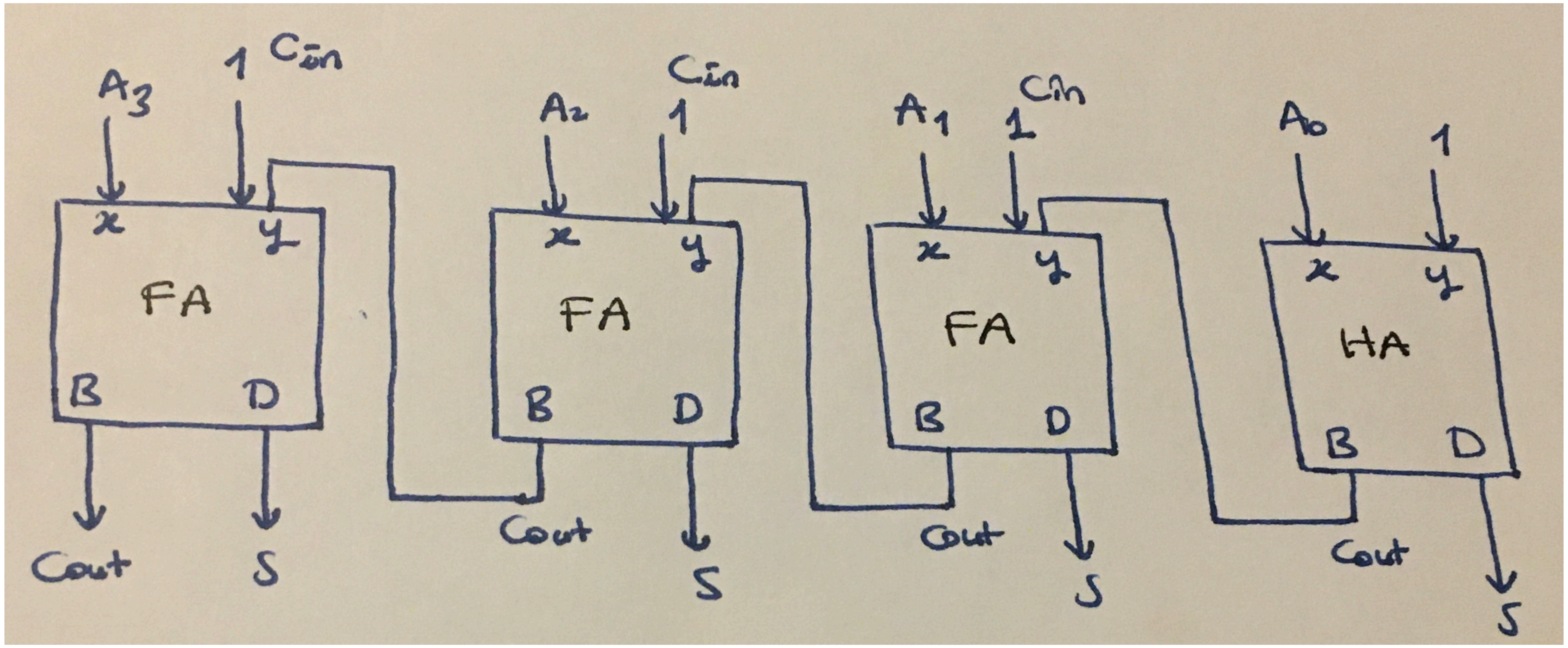
- Design a four-bit combinational circuit incrementer — A circuit that adds 1 to a four-bit binary number.



4-bit incrementer

Exercise 03: Full-Adder

- Design a four-bit combinational circuit decrementer — A circuit that subtracts 1 from a four-bit binary number.

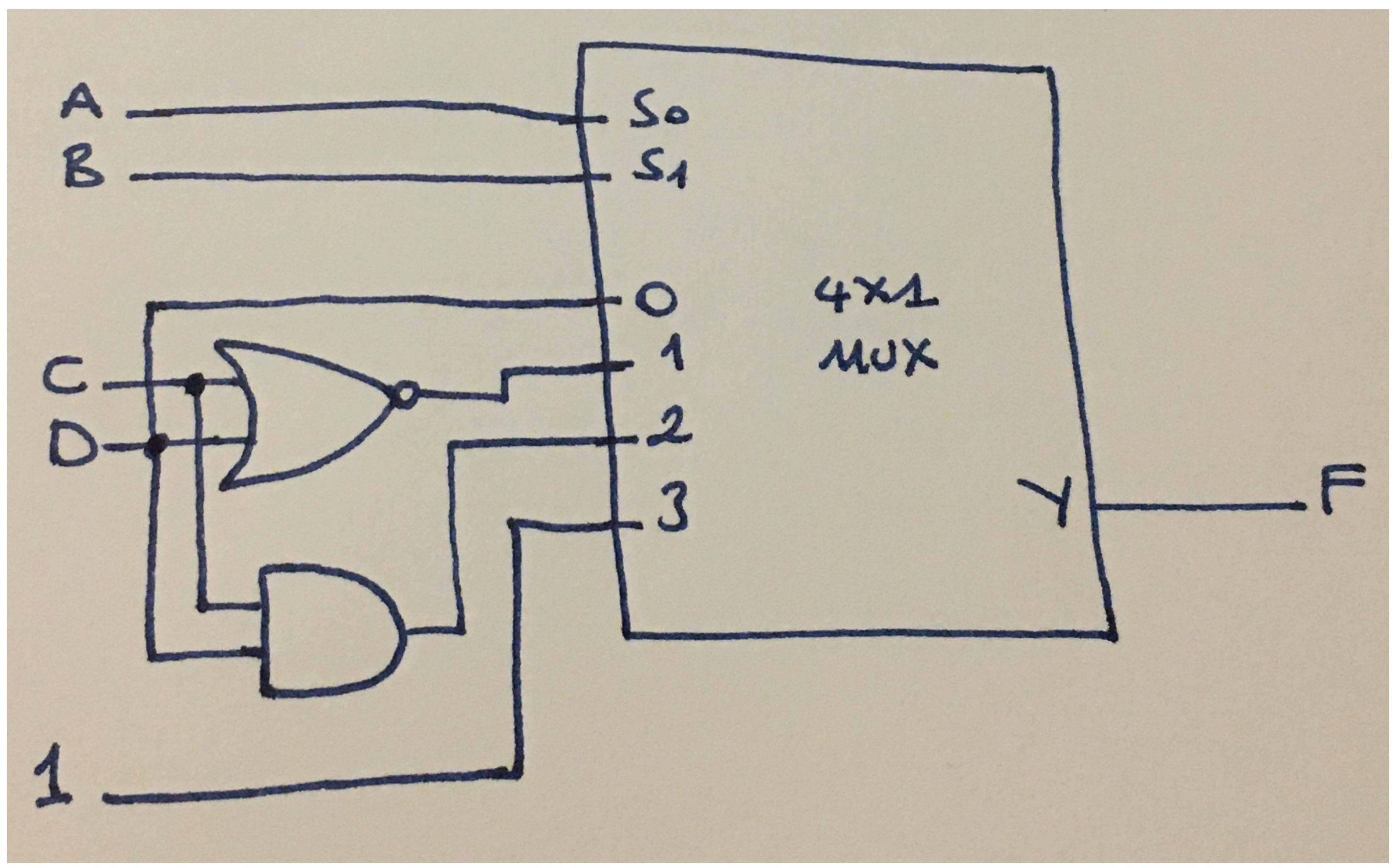


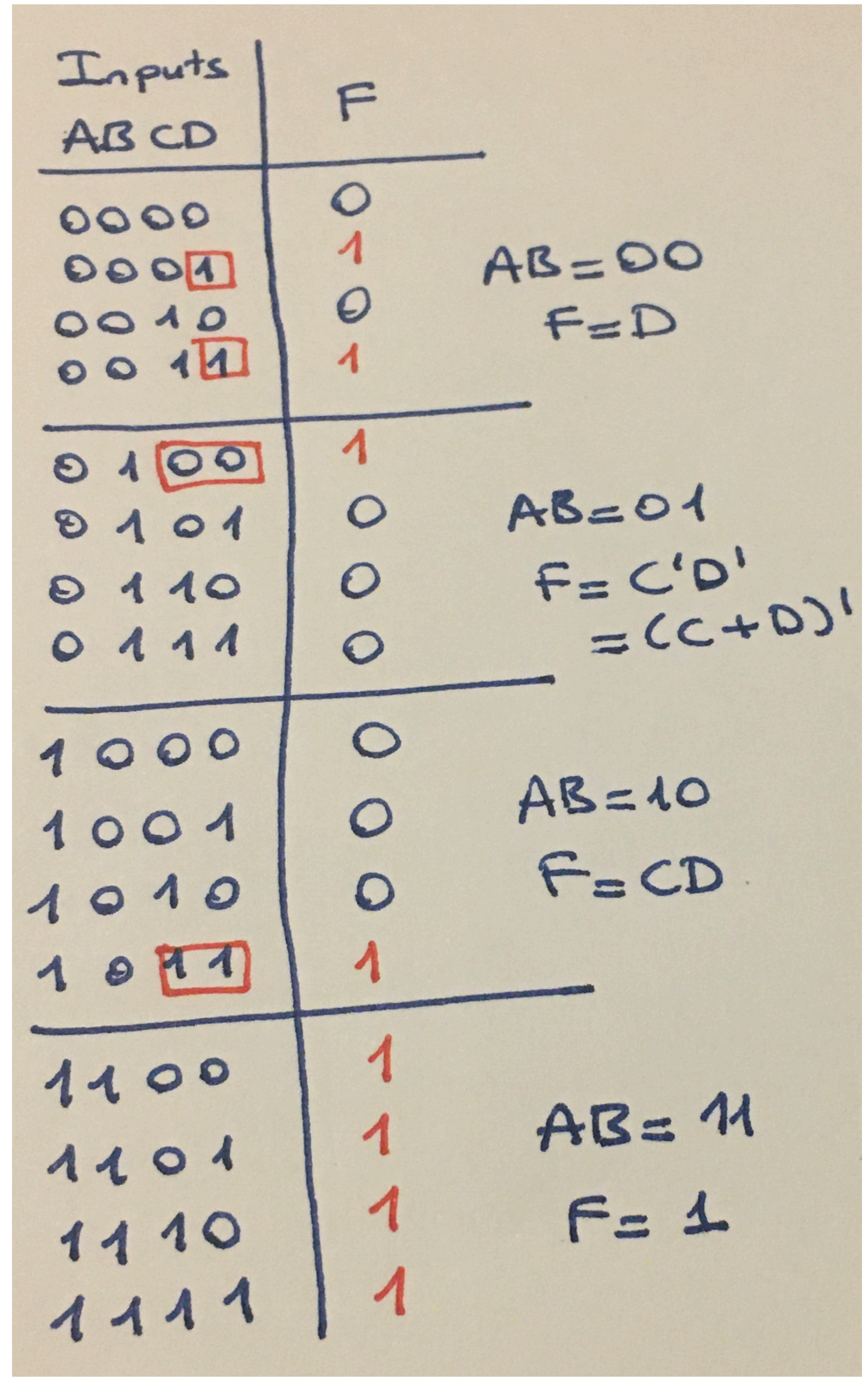
4-bit decrementer

Exercise 04: Multiplexer

Implement the following Boolean function with a 4x1 multiplexer and logic gates.

- `F(A, B, C, D) = Σ (1, 3, 4, 11, 12, 13, 14, 15)`
- Connect inputs A and B to the selection lines.
- The input requirements for the four data lines will be a function of variables C and D.
- These values are obtained by expressing F as a function of C and C for each of the four cases when `AB = 00, 01, 10, 11.`
- The functions may have to be implemented with external gates and with connections to power and ground.





Thank you for time

Contact Info
E-mail: nese.gunes@itu.edu.tr