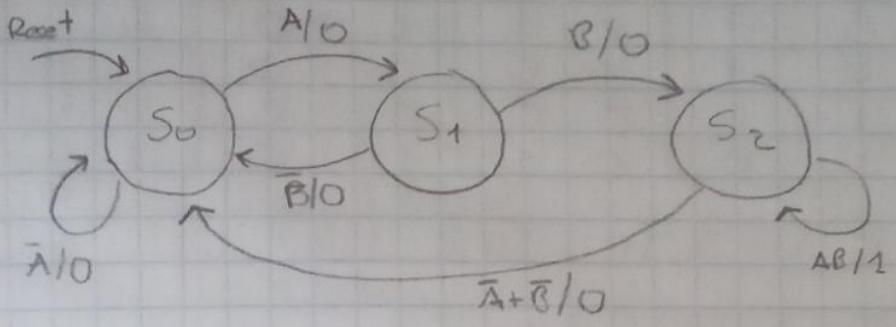
Laboratorio 06 Electrónica Digital 1

Ejercicio 1

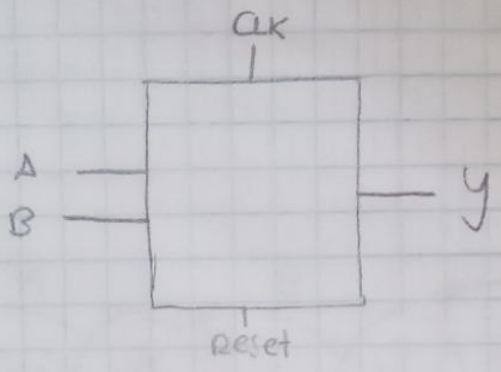
Fernando Arriaza

Lab 06 Digital I

I.



1.

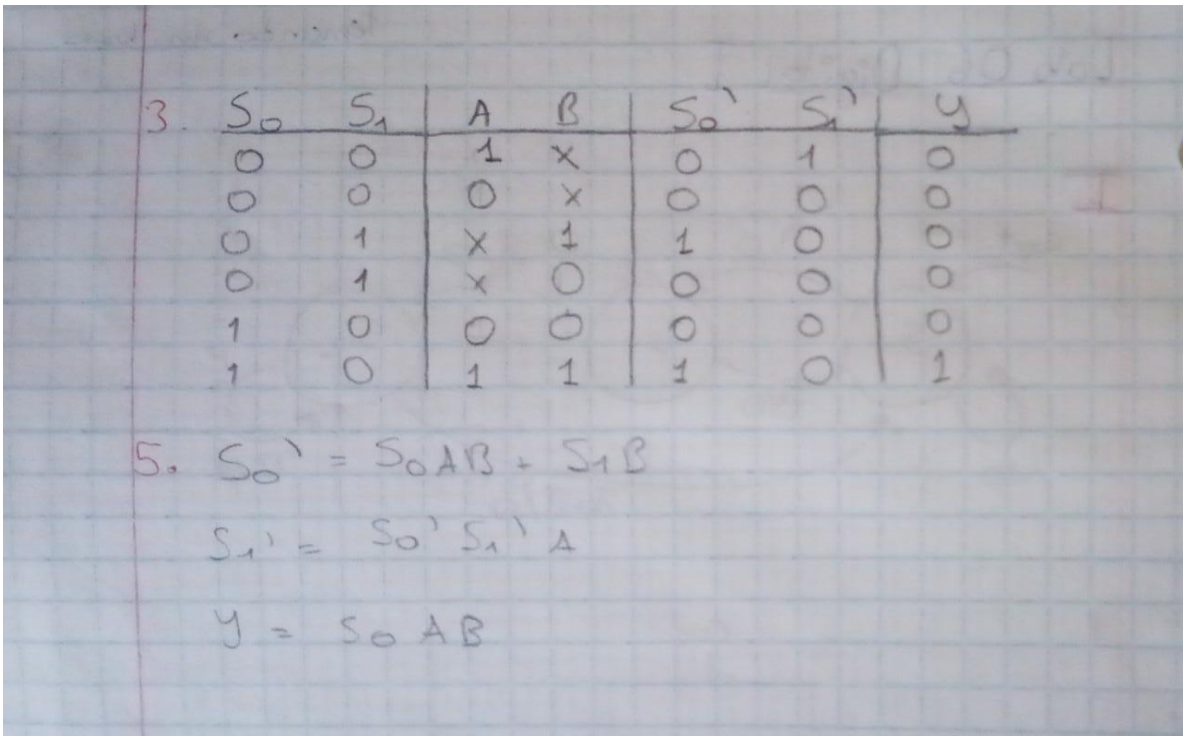


2.

S	A	B	S'	y
S_0	1	X	S_1'	0
S_0	0	X	S_0'	0
S_1	X	1	S_2'	0
S_1	X	0	S_0'	0
S_2	0	0	S_0'	0
S_2	1	1	S_2'	1

Codificación

00	S_0
01	S_1
10	S_2



3.

S_0	S_1	A	B	S_0'	S_1'	y
0	0	1	X	0	1	0
0	0	0	X	0	0	0
0	1	X	1	1	0	0
0	1	X	0	0	0	0
1	0	0	0	0	0	0
1	0	1	1	1	0	1

5. $S_0' = S_0AB + S_1B$

$S_1' = S_0' S_1' A$

$y = S_0 AB$

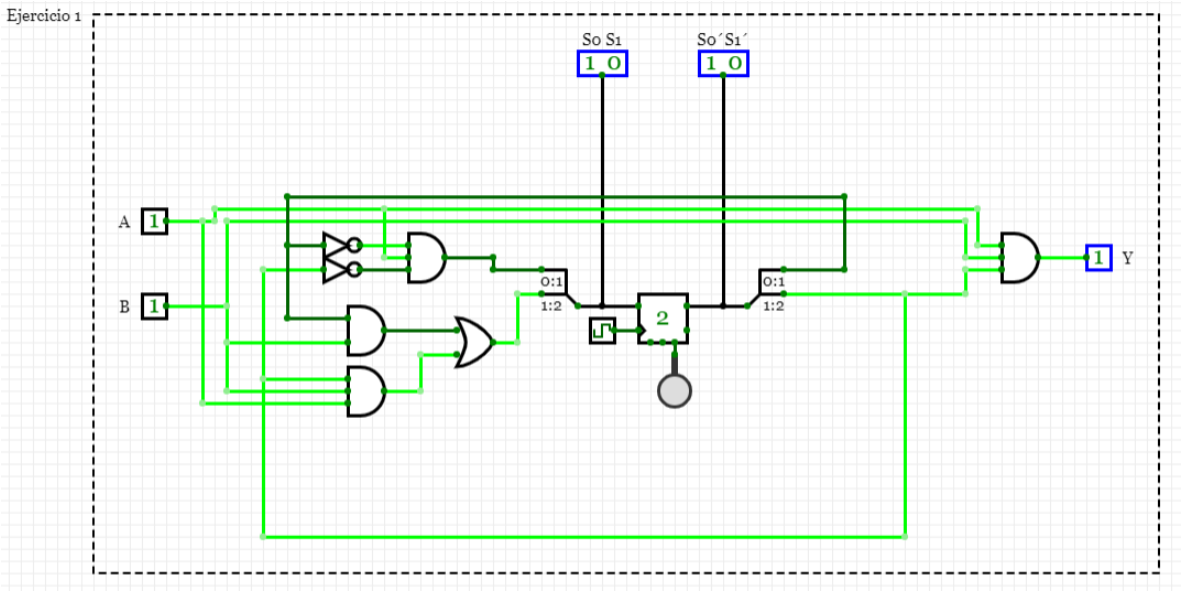
Logic Friday

File Operation Truthtable Equation Gates View Help

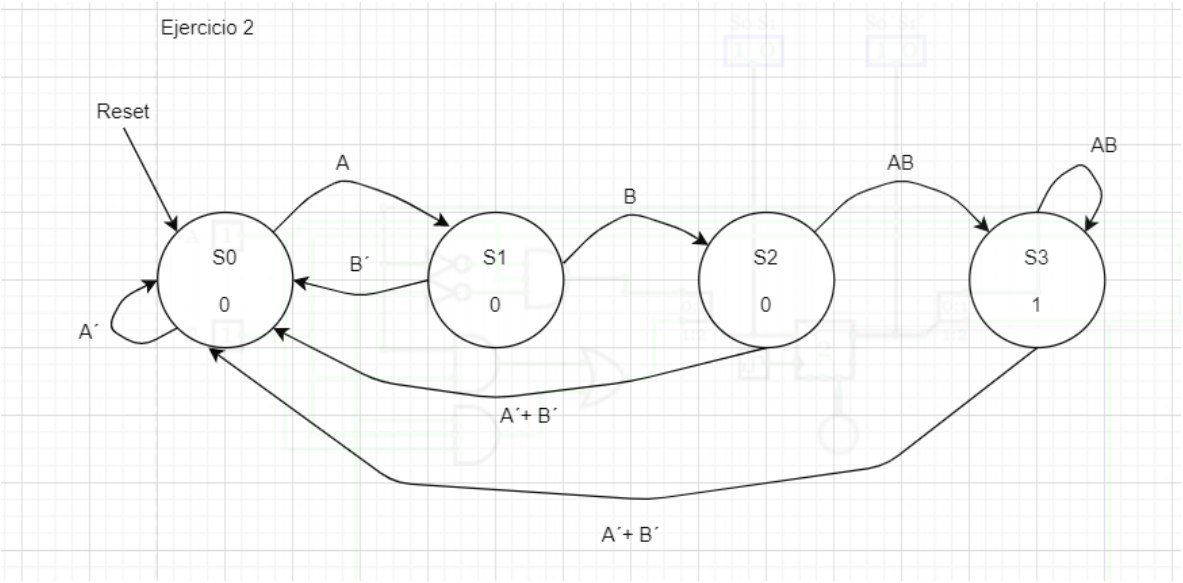
Functi...	Inputs	Outputs	True	False	DC	PI	Gates
S0'-Y	4	3	3, 2, 1	9, 10,...	4, 4, 4	3	9

S0	S1	A	B	=>	S0'	S1'	Y
1	X	1	1		1		1
X	1	X	1		1		
0	0	1	X			1	

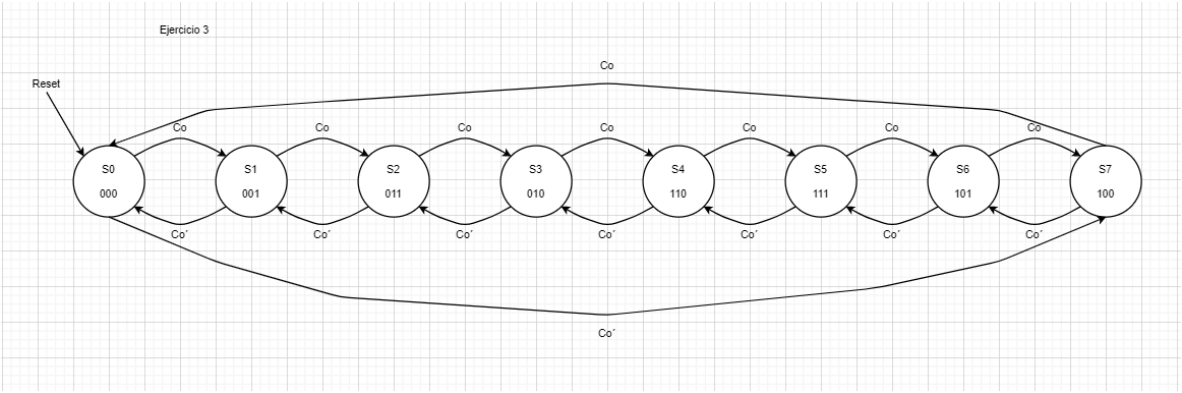
```
S0' = S0' S1 A' B + S0' S1 A B + S0 S1' A B;
S1' = S0' S1' A B' + S0' S1' A B;
Y = S0 S1' A B;

Minimized:
S0' = S0 A B + S1 B;
S1' = S0' S1' A ;
Y = S0 A B;
```


Ejercicio 1
So S1
1 0
So´S1´
1 0
A
1
B
1
0:1
1:2
2
0:1
1:2
1
Y

Ejercicio 2


Ejercicio 2
Reset
A
B´
B
AB
AB
A´
S0
0
S1
0
S2
0
S3
1
A´+ B´
A´+ B´

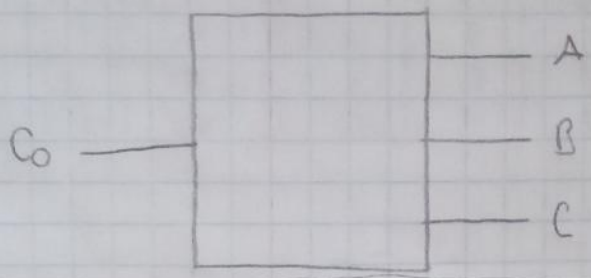
Ejercicio 3


Ejercicio 3
Reset
Co
Co
Co
Co
Co
Co
Co
Co
Co
S0
000
S1
001
S2
011
S3
010
S4
110
S5
111
S6
101
S7
100
Co´
Co´
Co´
Co´
Co´
Co´
Co´
Co´

III

No.	Cód. Gray
0	0 0 0
1	0 0 1
2	0 1 1
3	0 1 0
4	1 1 0
5	1 1 1
6	1 0 1
7	1 0 0

2.



3.1

Estado	Inputs	Salidas
S_0	0	S_7
S_0	1	S_1
S_1	0	S_0
S_1	1	S_2
S_2	0	S_1
S_2	1	S_3
S_3	0	S_2
S_3	1	S_4
S_4	0	S_3
S_4	1	S_5
S_5	0	S_4
S_5	1	S_6
S_6	0	S_5
S_6	1	S_7
S_7	0	S_6
S_7	1	S_0

	S_2 S_1 S_0
S_0	0 0 0
S_1	0 0 1
S_2	0 1 0
S_3	0 1 1
S_4	1 0 0
S_5	1 0 1
S_6	1 1 0
S_7	1 1 1

4.1

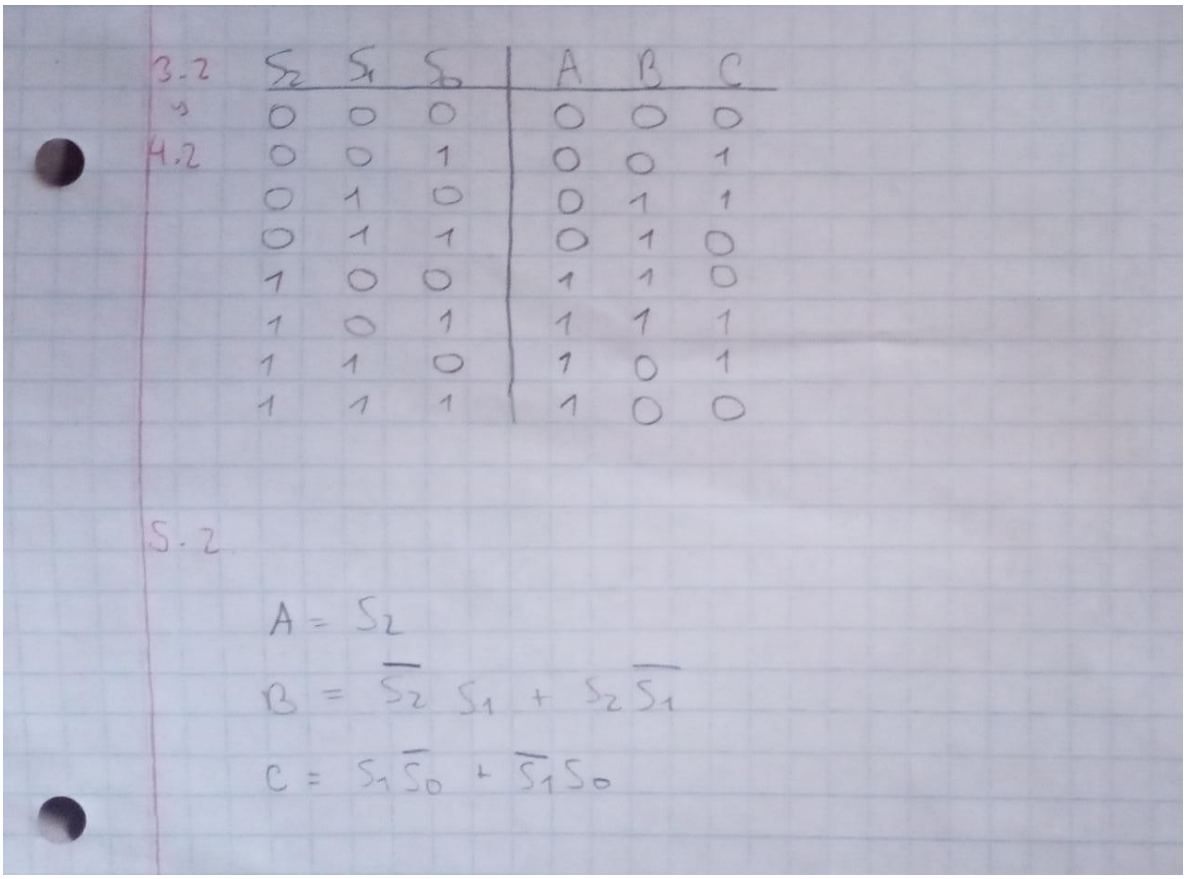
S_2	S_1	S_0	C_0	S_2'	S_1'	S_0'
0	0	0	0	1	1	1
0	0	0	1	0	0	1
0	0	1	0	0	0	0
0	0	1	1	0	1	0
0	1	0	0	0	0	1
0	1	0	1	0	1	1
0	1	1	0	0	1	0
0	1	1	1	1	0	0
1	0	0	0	0	1	1
1	0	0	1	1	0	1
1	0	1	0	1	0	0
1	0	1	1	1	1	0
1	1	0	0	1	0	1
1	1	0	1	1	1	1
1	1	1	0	1	1	0
1	1	1	1	0	0	0

5.1

$$S_2' = \bar{S_2} S_1 S_0 C_0 + \bar{S_2}\,\bar{S_1}\,\bar{S_0}\,\bar{C_0} + S_2 S_0 \bar{C_0} + S_2 \bar{S_1} C_0 + S_2 S_1 \bar{S_0}$$

$$S_1' = S_1 S_0 \bar{C_0} + \bar{S_1} S_0 C_0 + \bar{S_1}\,\bar{S_0}\,\bar{C_0} + S_1 \bar{S_0} C_0$$

$$S_0' = \bar{S_0}$$



3.2

S_2	S_1	S_0	A	B	C
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

4.2

5.2

$$A = S_2$$

$$B = \overline{S_2}\, S_1 + S_2\, \overline{S_1}$$

$$C = S_1 \overline{S_0} + \overline{S_1} S_0$$

Logic Friday

File Operation Truthtable Equation Gates View Help

Functi...	Inputs	Outputs	True	False	DC	PI	Gates
S2'-S0'	4	3	8, 8, 8	8, 8, 8	0, 0, 0	10	Not mapped

S2	S1	S0	Co	=>	S2'	S1'	S0'
0	1	1	1		1		
0	0	0	0		1		
1	X	1	0		1		
1	0	X	1		1		
X	1	1	0			1	
X	0	1	1			1	
1	1	0	X		1		
X	0	0	0			1	
X	1	0	1			1	
X	X	0	X				1

```
Entered by truthtable:
S2' = S2' S1' S0' Co' + S2' S1 S0 Co + S2 S1' S0' Co + S2 S1' S0 Co' + S2 S1' S0 Co
+ S2 S1 S0' Co' + S2 S1 S0' Co + S2 S1 S0 Co';
S1' = S2' S1' S0' Co' + S2' S1' S0 Co + S2' S1 S0' Co + S2' S1 S0 Co' + S2 S1' S0'
Co' + S2 S1' S0 Co + S2 S1 S0' Co + S2 S1 S0 Co';
S0' = S2' S1' S0' Co' + S2' S1' S0' Co + S2' S1 S0' Co' + S2' S1 S0' Co + S2 S1'
S0' Co' + S2 S1' S0' Co + S2 S1 S0' Co' + S2 S1 S0' Co;

Minimized:
S2' = S2' S1 S0 Co + S2' S1' S0' Co' + S2 S0 Co' + S2 S1' Co + S2 S1 S0' ;
S1' = S1 S0 Co' + S1' S0 Co + S1' S0' Co' + S1 S0' Co;
S0' = S0' ;
```

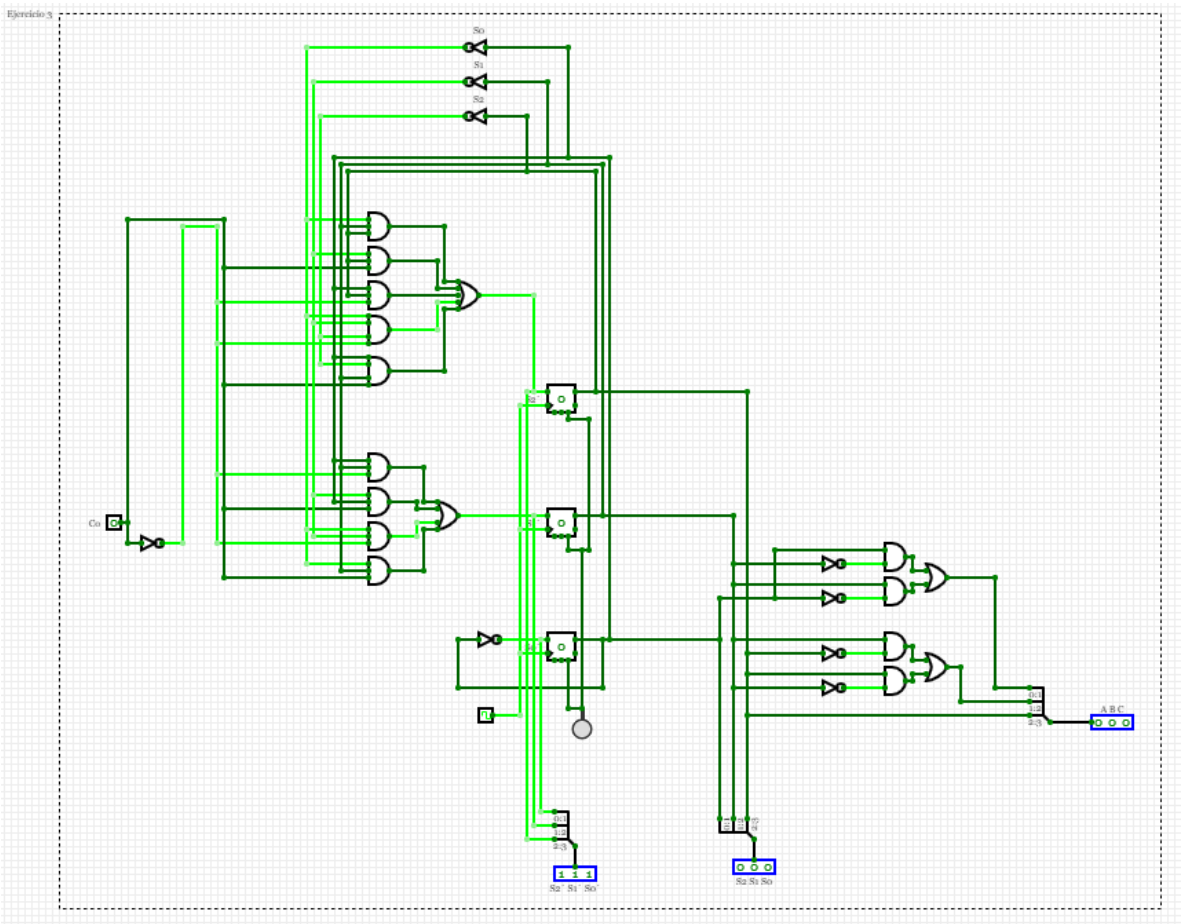
Logic Friday

File Operation Truthtable Equation Gates View Help

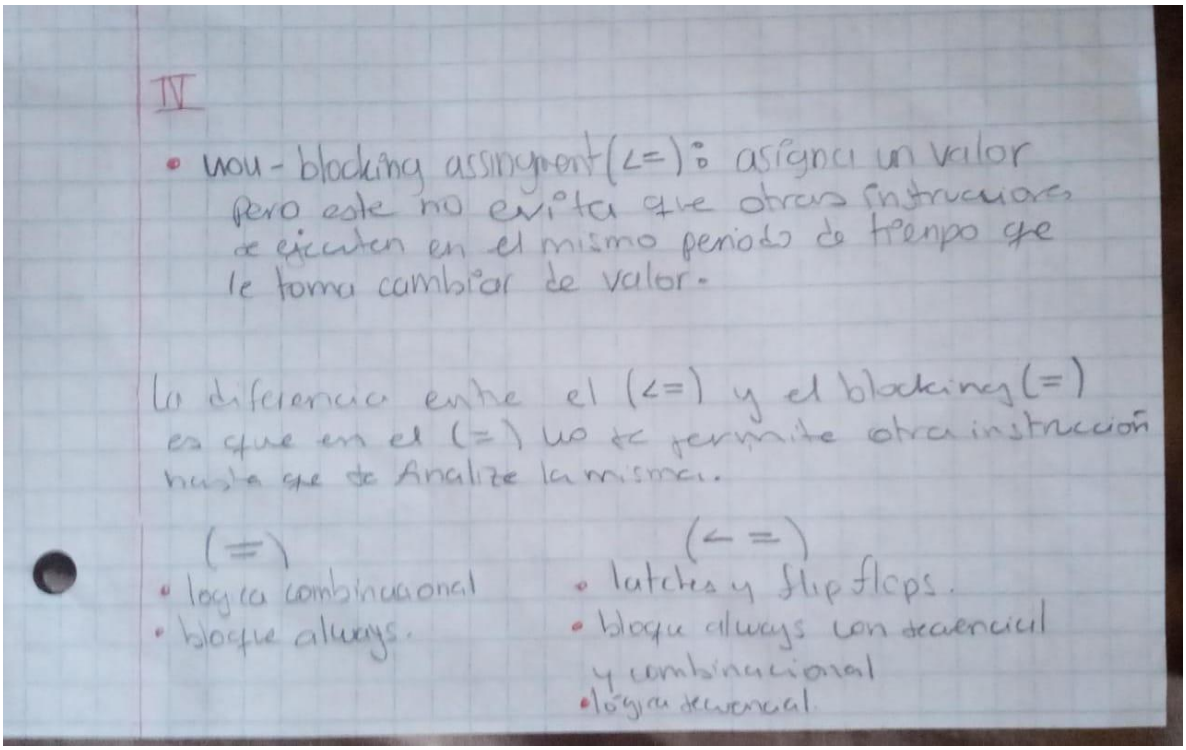
Functi...	Inputs	Outputs	True	False	DC	PI	Gates
A-C	3	3	4, 4, 4	4, 4, 4	0, 0, 0	5	Not mapped

S2	S1	S0	=>	A	B	C
0	1	X			1	
X	1	0				1
X	0	1				1
1	0	X			1	
1	X	X		1		

```
Entered by truthtable:
A = S2 S1' S0' + S2 S1' S0 + S2 S1 S0' + S2 S1 S0;
B = S2' S1 S0' + S2' S1 S0 + S2 S1' S0' + S2 S1' S0;
C = S2' S1' S0 + S2' S1 S0' + S2 S1' S0 + S2 S1 S0';

Minimized:
A = S2 ;
B = S2' S1  + S2 S1' ;
C = S1 S0' + S1' S0;
```



Ejercicio 4



Ejemplo Non-Blocking Assingment

```
module flipflop_4b(input logic clk, input logic reset, input logic en, input logic [3:0] d, output logic [3:0] q);

  always @ (posedge clk, posedge reset)
  if (reset) q <= 4'b0;
  else if (en) q <= d;

endmodule
```

Ejemplo Blocking Assignment

```
module tabla1(input wire A, B, C, output wire Y);

  assign Y = (~A & ~C) | (A & ~B) | (A & C);

endmodule
```