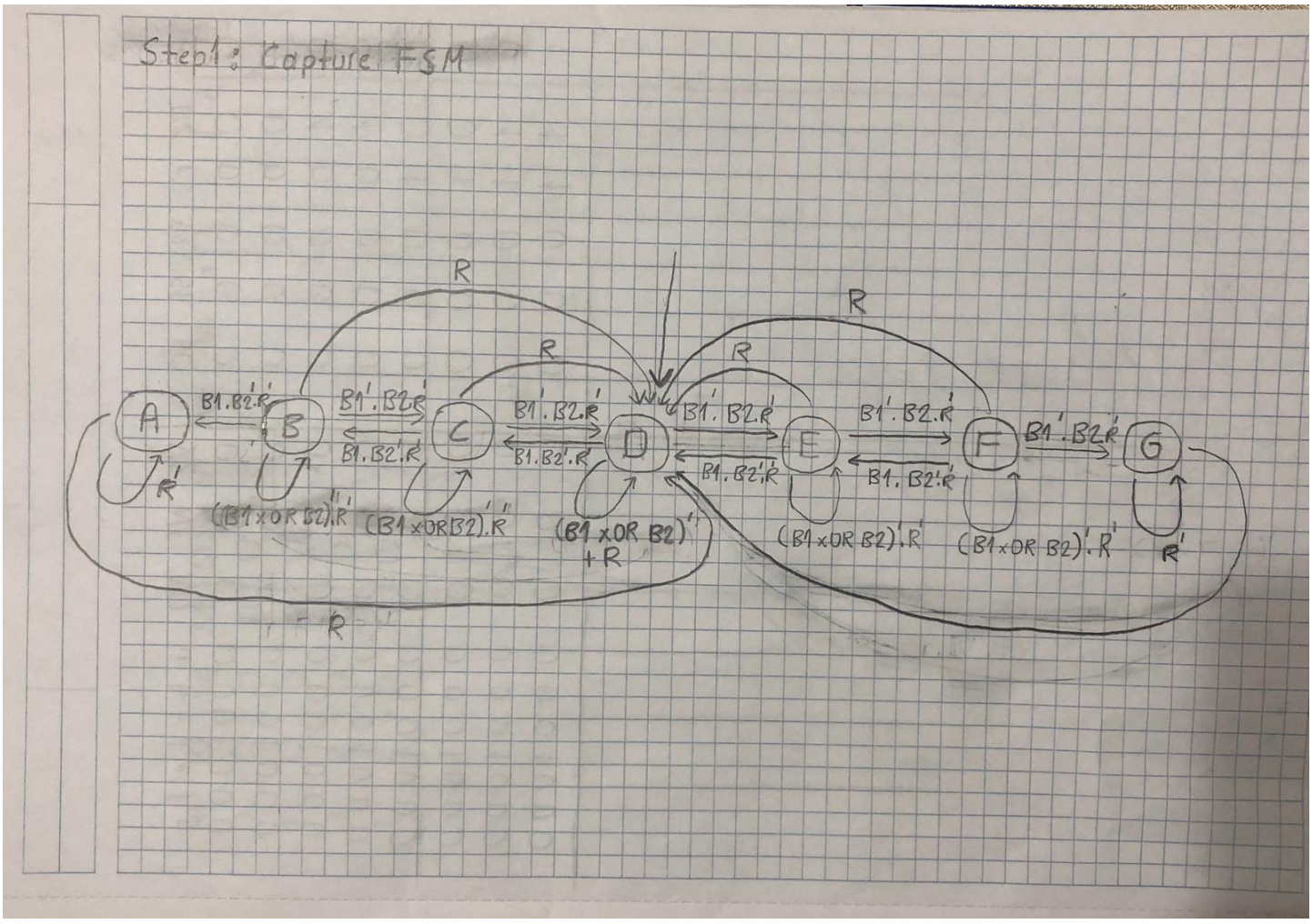
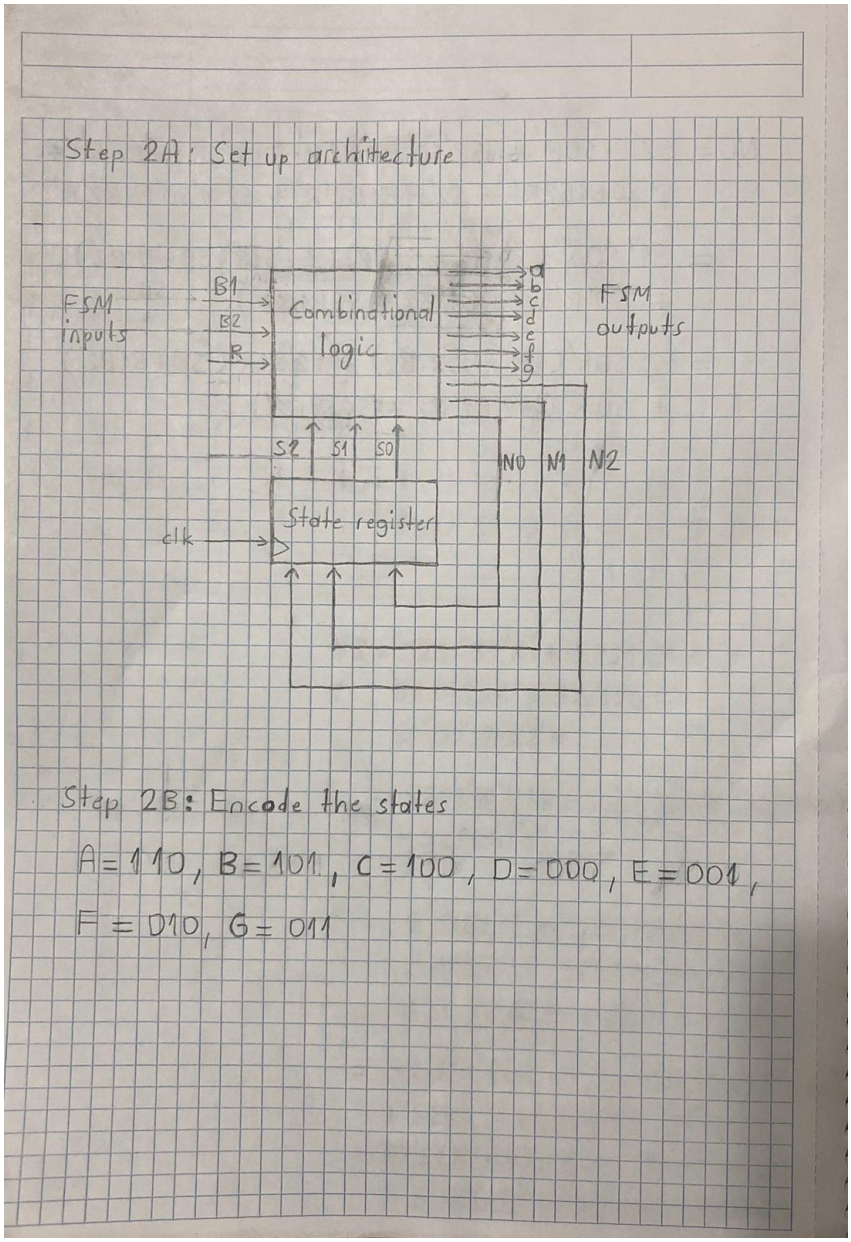
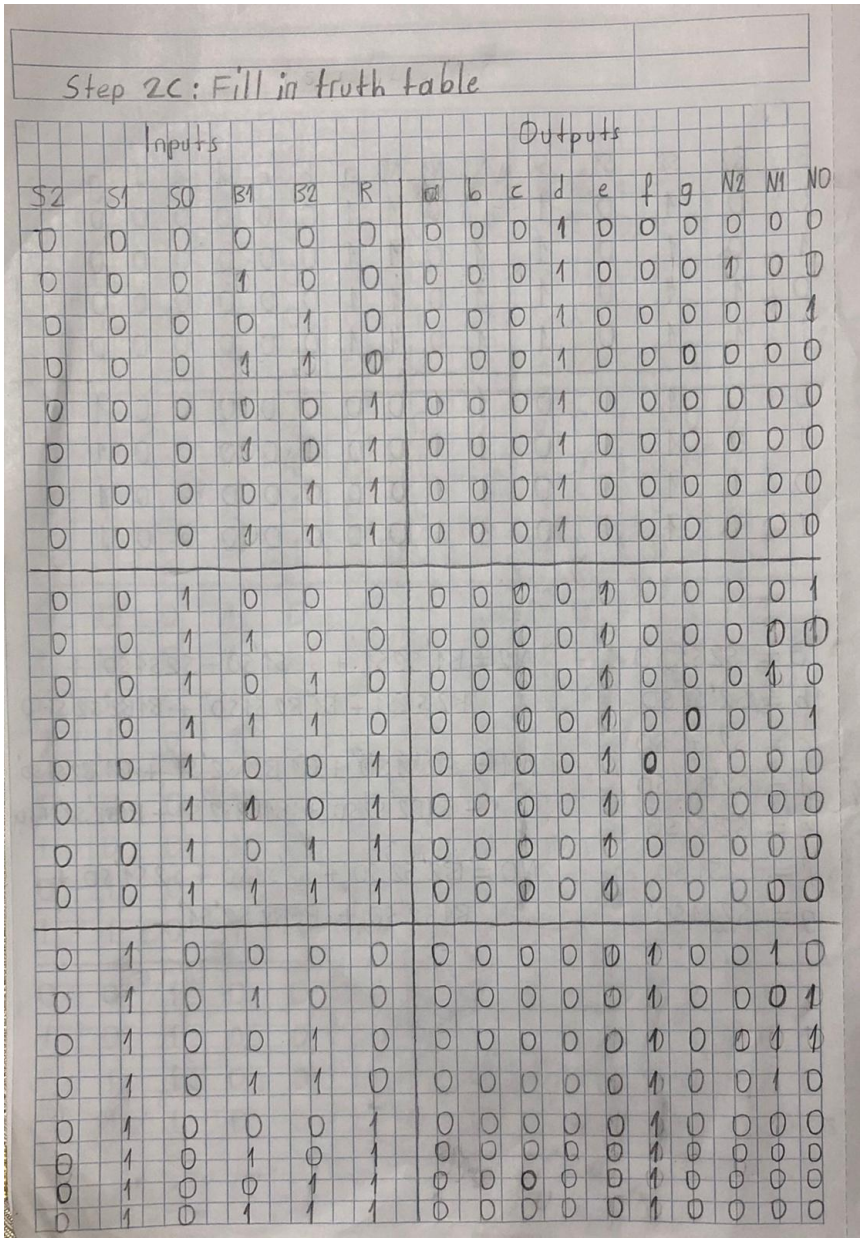

Step1: Capture FSM
A
B
C
D
E
F
G
R
B1.B2.R'
B1'.B2.R'
B1.B2'.R'
B1'.B2'.R
(B1 xOR B2)'.R'
(B1 xOR B2)'
+R

Step 2A: Set up architecture



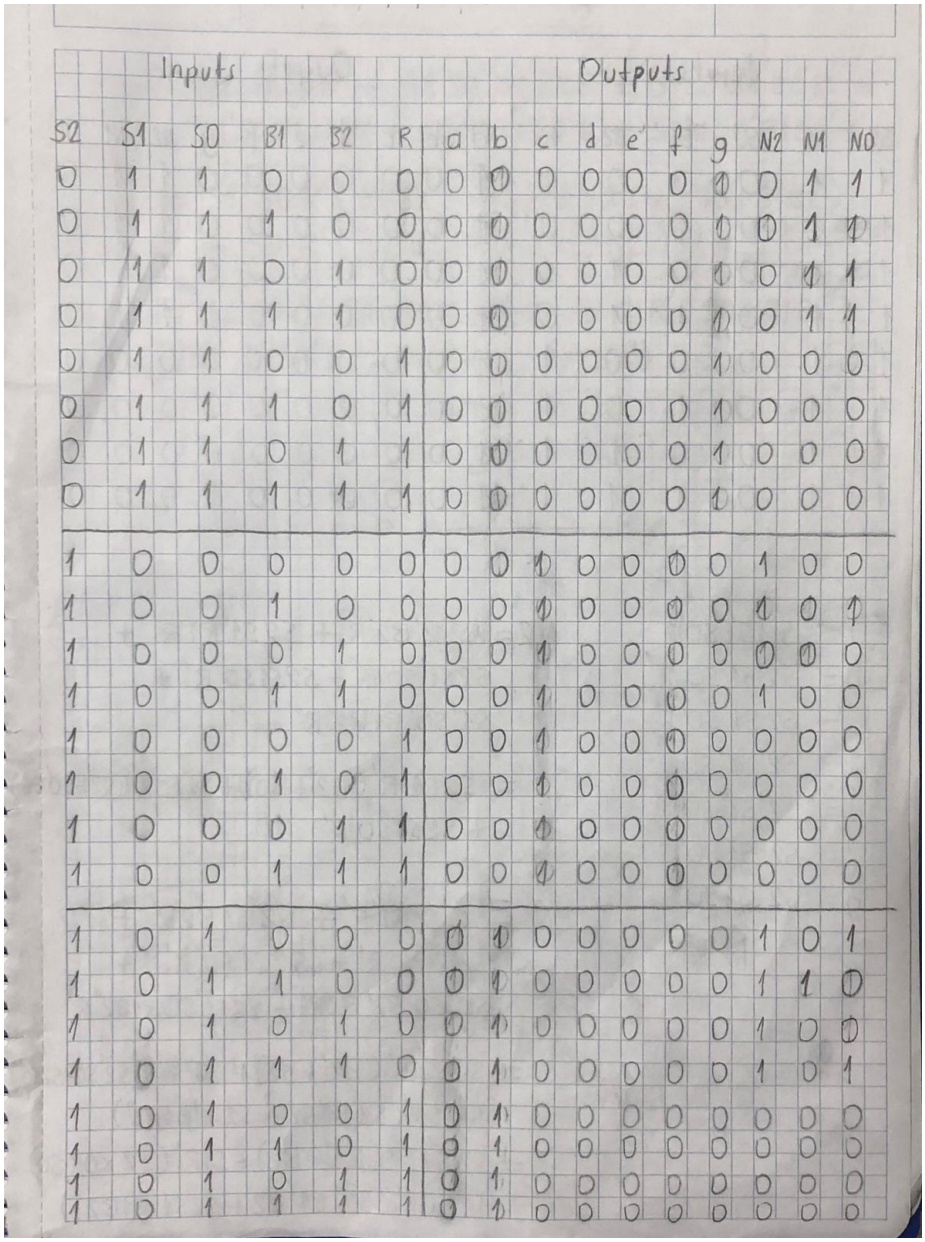
Step 2B: Encode the states

A = 110, B = 101, C = 100, D = 000, E = 001,
F = 010, G = 011



Step 2c: Fill in truth table

Inputs						Outputs									
S2	S1	S0	B1	B2	R	a	b	c	d	e	f	g	N2	N1	N0
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
0	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	1	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	0	0	1	1	1	0	0	0	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	1	0	0	0	1	0
0	0	1	1	1	0	0	0	0	0	1	0	0	0	0	1
0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0
0	0	1	1	0	1	0	0	0	0	1	0	0	0	0	0
0	0	1	0	1	1	0	0	0	0	1	0	0	0	0	0
0	0	1	1	1	1	0	0	0	0	1	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0
0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	1
0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	1
0	1	0	1	1	0	0	0	0	0	0	1	0	0	1	0
0	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	0	0	1	1	0	0	0	0	0	1	0	0	0	0
0	1	0	1	1	1	0	0	0	0	0	1	0	0	0	0



Inputs						Outputs									
S2	S1	S0	B1	B2	R	a	b	c	d	e	f	g	N2	N1	N0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1
0	1	1	1	0	0	0	0	0	0	0	0	1	0	1	1
0	1	1	0	1	0	0	0	0	0	0	0	1	0	1	1
0	1	1	1	1	0	0	0	0	0	0	0	1	0	1	1
0	1	1	0	0	1	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	1	0	0	0	0	0	0	1	0	0	0
0	1	1	0	1	1	0	0	0	0	0	0	1	0	0	0
0	1	1	1	1	1	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0
1	0	0	1	0	0	0	0	1	0	0	0	0	1	0	1
1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	1	0	0	0	0	1	0	0
1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0
1	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0
1	0	0	1	1	1	0	0	1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	1	0	0	0	0	0	1	0	1
1	0	1	1	0	0	0	1	0	0	0	0	0	1	1	0
1	0	1	0	1	0	0	1	0	0	0	0	0	1	0	0
1	0	1	1	1	0	0	1	0	0	0	0	0	1	0	1
1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0
1	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0
1	0	1	0	1	1	0	1	0	0	0	0	0	0	0	0
1	0	1	1	1	1	0	1	0	0	0	0	0	0	0	0

Inputs						Outputs									
S2	S1	S0	B1	B2	R	a	b	c	d	e	f	g	N2	N1	N0
1	1	0	0	0	0	1	0	0	0	0	0	0	1	1	0
1	1	0	1	0	0	1	0	0	0	0	0	0	1	1	0
1	1	0	0	1	0	1	0	0	0	0	0	0	1	1	0
1	1	0	1	1	0	1	0	0	0	0	0	0	1	1	0
1	1	0	0	0	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	0	1	1	0	0	0	0	0	0	0	0	0
1	1	0	0	1	1	1	0	0	0	0	0	0	0	0	0
1	1	0	1	1	1	1	0	0	0	0	0	0	0	0	0

CamScanner ile tarandı

Inputs						Outputs	
S2	S1	S0	B1	B2	R	User1	User2
1	1	0	0	0	0	1	0
1	1	0	1	0	0	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	0
1	1	0	0	0	1	1	0
1	1	0	1	0	1	1	0
1	1	0	0	1	1	1	0
1	1	0	1	1	1	1	0
0	1	1	0	0	0	0	1
0	1	1	1	0	0	0	1
0	1	1	0	1	0	0	1
0	1	1	1	1	0	0	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	1
0	1	1	0	1	1	0	1
0	1	1	1	1	1	0	1

CamScanner ile tarandı

a = S2 S1 S0'

b = S2 S1' S0

c = S2 S1' S0'

d = S2' S1' S0'

e = S2' S1' S0

f = S2' S1 S0'

g = S2' S1 S0

$N2 = S2S1'B2'R' + S2S1'B1R' + S2S1'S0R' + S2S1S0'R' + S1'S0'B1B2'R'$

$= S2S1'R'(B2' + S0) + S1'B1R'(S2 + S0'B2') + S2S1S0'R'$

$N1 = S2'S1B1'R' + S2'S1B2R' + S2'S1S0R' + S2S1S0'R' + S2'S0B1'B2R' + S2S1'S0B1B2'R'$

$= S2'S1R'(B1' + B2) + S1R'(S2'S0 + S2S0') + S0R'(S2'B1'B2 + S2S1'B1B2')$

$= S2'S1R'(B1' + B2) + S1R'(S2 \times OR\ S0) + S0R'(S2'B1'B2 + S2S1'B1B2')$

CamScanner ile tarandı

$$NO = S2'S1SOR' + S2'SO'B1'B2R' + S1'SO'B1'B2'R' + S1'SOB1B2R' + S2'S1B1B2'R' + S2S1'SO'B1B2'R'$$

$$= S1'SOR'(B1'B2' + B1B2) + S2'S1R'(SO + B1B2') + SO'R'(S2'B1'B2 + S2B1B2')$$

$$= S1'SOR'(B1 \text{ xNOR } B2) + S2'S1R'(SO + B1B2') + SO'R'(S2'B1'B2 + S2B1B2')$$

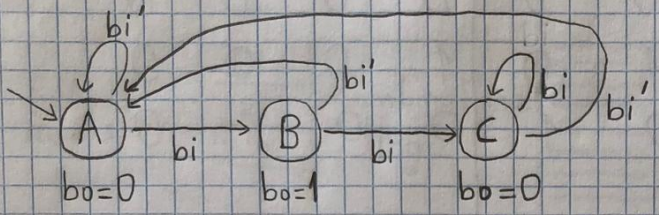
$$User1 = S2S1SO'$$

$$User2 = S2'S1SO$$

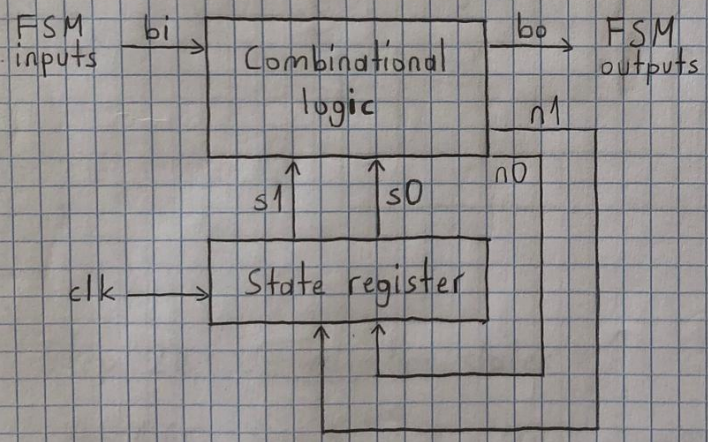
CamScanner ile tarandı

Button Press Synchronizer

Step 1: Capture FSM



Step 2A: Set up architecture

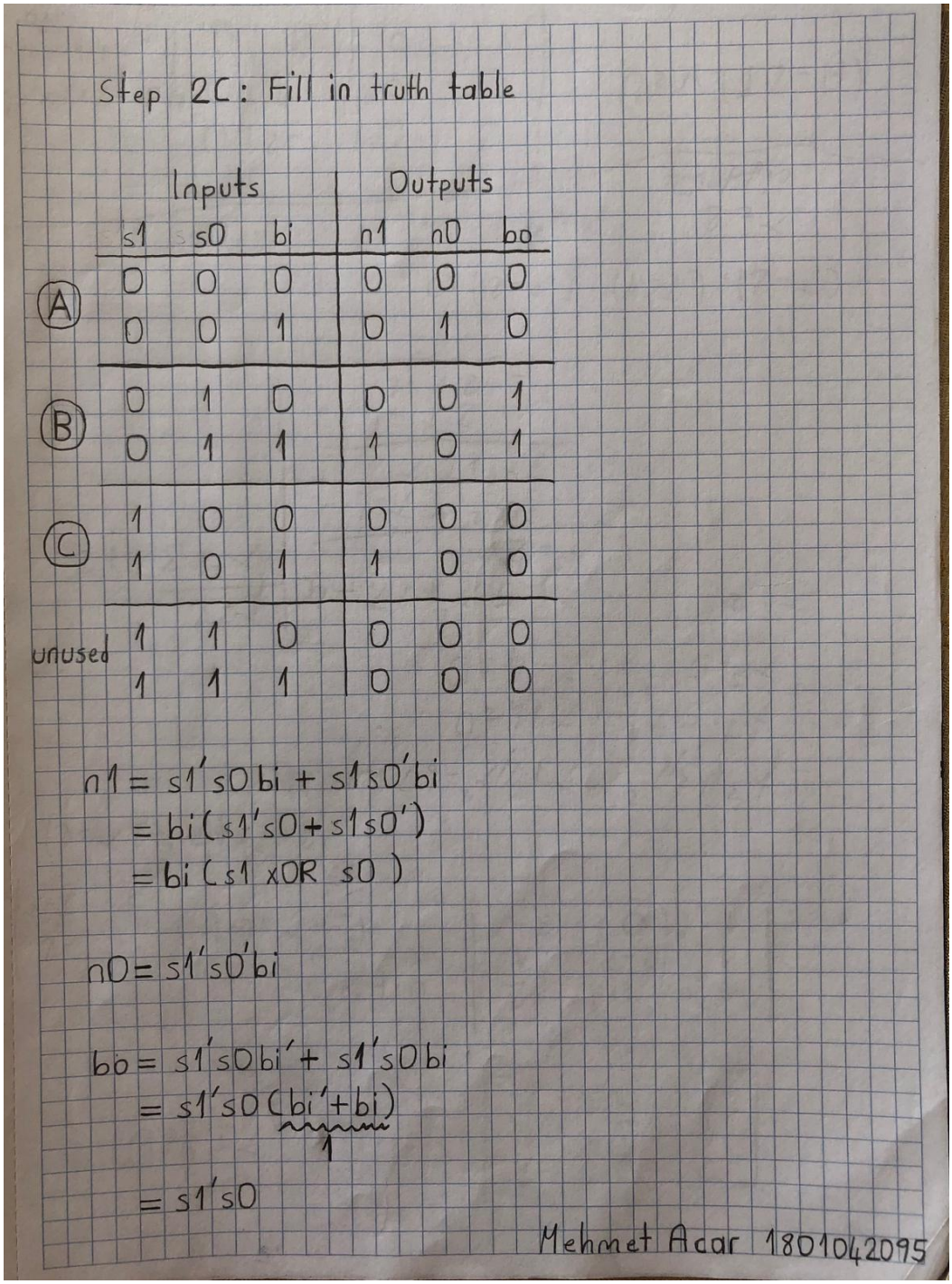


Step 2B: Encode states

A = 00

B = 01

C = 10



Step 2C: Fill in truth table

	Inputs			Outputs		
	s1	s0	bi	n1	n0	bo
A	0	0	0	0	0	0
	0	0	1	0	1	0
B	0	1	0	0	0	1
	0	1	1	1	0	1
C	1	0	0	0	0	0
	1	0	1	1	0	0
unused	1	1	0	0	0	0
	1	1	1	0	0	0

$n1 = s1's0bi + s1s0'bi$

$= bi(s1's0 + s1s0')$

$= bi(s1 \text{ xOR } s0)$

$n0 = s1's0'bi$

$bo = s1's0bi' + s1's0bi$

$= s1's0(bi' + bi)$

$(bi' + bi) = 1$

$= s1's0$

Mehmet Acar 1801042095